Disentangled Dynamic Intrusion Detection

Chenyang Qiu, Guoshun Nan, *Member, IEEE,* Hongrui Xia, Zheng Weng, Xueting Wang,
Meng Shen, *Member, IEEE,* Xiaofeng Tao, *Senior Member, IEEE*, Jun Liu

Abstract—Network-based intrusion detection system (NIDS) monitors network traffic for malicious activities, forming the frontline defense against increasing attacks over information infrastructures. Although promising, our quantitative analysis shows that existing methods perform inconsistently in attacks (e.g., 18% F1 for the MITM and 93% F1 for DDoS by a GCN-based state-of-the-art method), and perform poorly in few-shot intrusion detections (e.g., dramatically drops from 91% to 36% in 3D-IDS, and drops from 89% to 20% in E-GraphSAGE). We reveal that the underlying cause is entangled distributions of flow features. This motivates us to propose DIDS-MFL, a disentangled intrusion detection approach for various scenarios. DIDS-MFL involves two key components: a double Disentanglement-based Intrusion Detection System (DIDS) and a plug-and-play Multi-scale Few-shot Learning-based (MFL) intrusion detection module. Specifically, the proposed DIDS first disentangles traffic features by a non-parameterized optimization, automatically differentiating tens and hundreds of complex features. Such differentiated features will be further disentangled to highlight the attack-specific features. Our DIDS additionally uses a novel graph diffusion method that dynamically fuses the network topology for spatial-temporal aggregation in evolving data streams. Furthermore, the proposed MFL involves an alternating optimization framework to address the entangled representations in few-shot traffic threats with rigorous derivation. MFL first captures multi-scale information in latent space to distinguish attack-specific information and then optimizes the disentanglement term to highlight the attack-specific information. Finally, MFL fuses and alternately solves them in an end-to-end way. To the best of our knowledge, DIDS-MFL takes the first step toward disentangled dynamic intrusion detection under various attack scenarios. Equipped with DIDS-MFL, administrators can effectively identify various attacks in encrypted traffic, including known, unknown, and few-shot threats that are not easily detected. Comprehensive experiments show the superiority of our proposed DIDS-MFL. For few-shot NIDS, our DIDS-MFL achieves a 71.91% - 125.19% improvement in average F1-score over 14 baselines and shows versatility in multiple baselines and multiple tasks. Our code is available at https://github.com/qcydm/DIDS-MFL

Index Terms—Intrusion Detection, Network Security.

I. INTRODUCTION

C. Qiu, G. Nan, H. Xia, Z. Weng, X. Wang, and X. Tao are with the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications, Beijing, 100876, China, and also with the Beiyou Shenzhen Institute, Shenzhen, China. E-mail: {cyqiu, nanguo2021, kphchrls, wengzheng, wxtyuki, taoxf} @bupt.edu.cn

M. Shen is with Beijing Institute of Technology, Beijing, 100081, China. E-mail: shenmeng@bit.edu.cn

J. Liu is with the School of Computing and Communications, Lancaster University, United Kingdom. E-mail: j.liu81@lancaster.ac.uk

Corresponding author: G. Nan (email: nanguo2021@bupt.edu.cn).

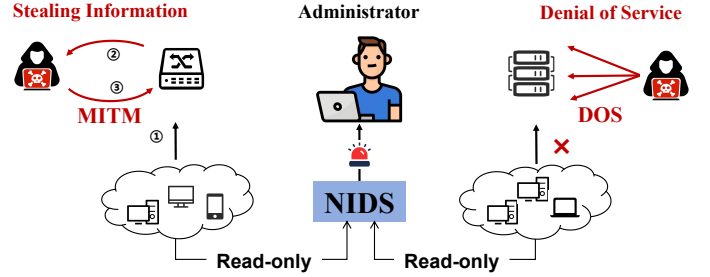Manuscript received April 19, 2021; revised August 16, 2021.

Fig. 1. Illustration of two network attacks DoS and MITM. DoS floods the target with massive traffic to overwhelm an online service, and MITM eavesdrops on the communication between two targets and steals private information. An NIDS can be easily deployed in a single location to collect statistical features and alert administrators for potential threats.

UNAUTHORIZED attempts like password cracking [1], man-in-the-middle attacks (MITM) [2], and denial-of-service (DoS) [3] are known as the network attacks [4], targeting an organization's digital assets with the intention of data leakage or the execution of harmful deeds. These attacks are frequently perceived as network anomalies [5] due to their distinct characteristics that differ from standard traffic patterns. An alarming statistic [6] notes that 31% companies across the globe experience a daily average of at least one cyber attack, a frequency amplified by the proliferation of mobile online business endeavors. The situations necessitate an intelligent system deployment to assist network administrators in the automated segregation of these anomalies from the vast sea of internet traffic. A network-based intrusion detection system (NIDS) [7], which monitors network traffic and identifies malicious activities, facilitates administrators to form the frontline defense against increasing attacks over information infrastructures (e.g., sensors and servers). Hence, NIDS is widely applied in many information systems of governments and e-commercial business sectors [8]. Figure 1 demonstrates how NIDS builds a frontline defense against two network attacks, protecting systems from potential cyber threats.

Existing NIDS can be categorized into two types, i.e., signature-based ones [9]–[11] and anomaly-based ones [12]–[14]. The former detects network attacks based on pre-defined patterns or known malicious sequences stored in a database, such as the number of bytes in traffic. These patterns in the NIDS are referred to as signatures. The latter anomaly-based NIDS learns to track attacks with machine learning techniques. Early statistical approaches [2], [3], such as Support Vector Machine (SVM), Logistic Regression (LR), and Decision Tree (DT), rely on carefully designed handcrafted features to learn classification boundaries. Recent deep learning-based methods [15], [16] use millions of neural parameters to mine
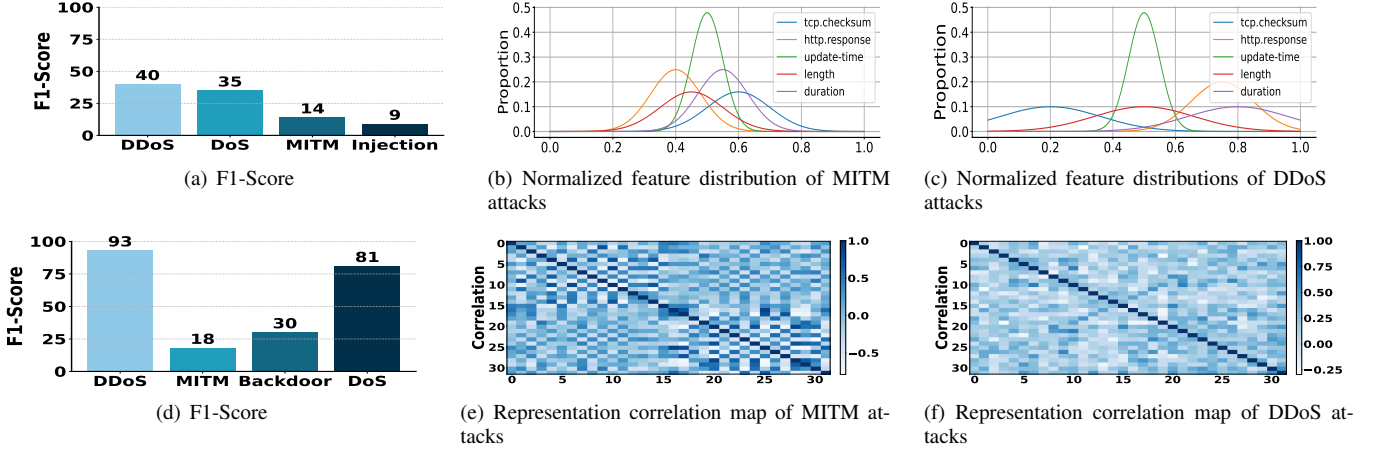
Fig. 2. Quantitative analysis on CIC-TON-IOT. (a) Comparisons of detecting various attacks, which are regarded as an unknown type in evaluation. Specifically, we train an SVM model without using the data points of these attacks, and evaluate the instances of these attacks on the test set. (b) and (c) show the feature distributions of two attacks, MITM and DDoS, respectively. (d) Comparisons of detecting various known attacks on the previous state-of-the-art deep learning model E-GraphSAGE, (e) and (f) are correlation maps of representations of the two attacks, where the representations are generated by E-GraphSAGE.
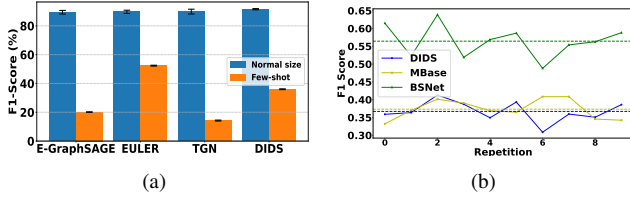

Fig. 3. (a) Performance comparisons among different intrusion detection methods with normal-size traffic and few-shot traffic. (b) F1-score comparisons between DIDS few-shot learning and two SOTA few-shot learning baselines. We repeat the above comparisons 10 times.

the knowledge underlying the training samples, and have achieved great success in automatically modeling complex correlations for tens and thousands of features. The state-of-the-art E-GraphSAGE [17] employs graph convolution networks (GCNs) to learn the feature representations for better prediction.

Although promising, existing NIDS approaches face two challenges in detecting traffic threats accurately:

- **Challenge** 1. **The existing NIDS methods yield extremely inconsistent results in identifying distinct attacks.** For statistical methods, Figure 2 (a) demonstrates that the detection performance of an SVM-based method [18] for an unknown attack [1] can be as low as 9% in terms of F1 on the CIC-ToN-IOT [19] dataset. While the model achieves 40% F1 in declaring another unknown threat (DDoS) on the same benchmark. Regarding the deep learning-based methods, Figure 2 (d) demonstrates that E-GraphSAGE achieves a lower than 20% F1 score for MITM attacks and a higher than 90% F1 score for DDoS on the CTC-ToN-IOT dataset.
- **Challenge** 2. **The existing NIDS approaches struggle to detect few-shot traffic threats accurately.** In practical scenarios, the strengthening of defensive measures and the consequent escalating trend of cyber threats have inevitably caused the rise of new-type attacks with a limited

[1]The unknown attacks referred to in this paper are never appeared new type of attacks in training set

instance number, referred to as few-shot traffic threats. However, our empirical observations reveal that the NIDS methods have dramatic performance deterioration in few-shot scenarios, as shown in Fig. 3 (a), e.g., our previous work DIDS [20] dramatically drops from 91.57% under supervised learning setting to 36.12% under the few-shot learning setting over the CIC-ToN-IoT dataset, and similarly, EURLER and E-GraphSAGE have an average F1-scores lower to 36.26% under the few-shot learning setting.

For the first challenge, we depict feature distributions and visualize the representations to investigate the underlying cause of why existing methods perform inconsistently for various attacks, including unknown ones and known ones. Figure 2 (b) and (c) depict statistical distributions of the two unknown attacks for the SVM-based model during testing. We observe that feature distributions of MITM attacks are entangled, while the ones of DDoS are more separated. It can be inferred that statistical distributions of traffic features are one of the main underlying causes of performance variations. Separated distributions benefit the unknown attack identification, while entangled ones are indistinguishable and unable to help the NIDS to make accurate decisions. We refer to such a phenomenon as **the entangled distribution of statistical features**. To analyze the reason for performance variations of acknowledged attacks during testing, we use Pearson correlation heat map [21] to visualize the representations of MITM and DDoS respectively, where the representations are generated by the encoder of E-GraphSAGE. Figure 2 (e) and Figure 2 (f) demonstrate the two correlation maps. Interestingly, we observe that the coefficients of MITM representations are much larger than those of DDoS. We further compare MITM with other attacks, including Backdoor and Dos, and find those high coefficients in the representation will lead to lower intrusion detection scores. We refer to such a phenomenon as **the entangled distribution of representational features**, which can be considered as another main cause for the degradation of attack classification.

For the second challenge, we observe that the NIDS ap-

(a) The t-SNE of DIDS

(b) The t-SNE of BSNet

(c) The correlation map of DIDS

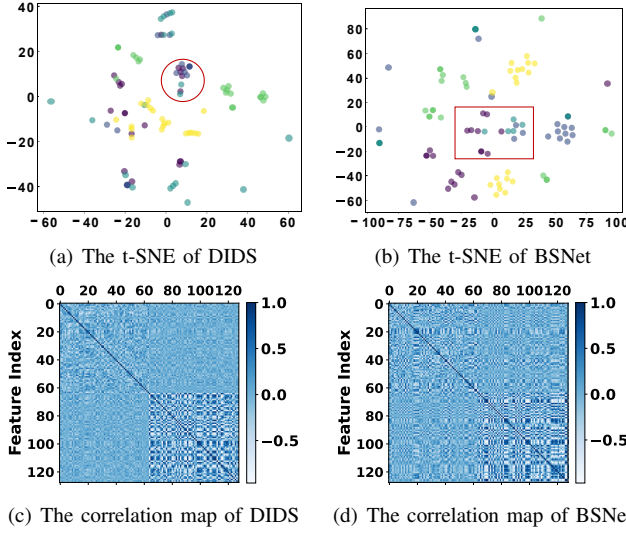(d) The correlation map of BSNet

Fig. 4. The correlation map and the t-SNE visualization of representations generated by DIDS and BSNet on CIC-ToN-IoT dataset.

proaches still perform poorly when combining the state-of-the-art few-shot learning methods, as shown in Fig. 3 (b). For the meta-learning-based MBase [22], we attribute the poor performance to the limited anomaly types in network traffic. Effective meta-learning approaches typically necessitate vast types of few-shot anomalies, which are more aligned with the field of computer vision and natural language processing [23]–[26]. Therefore, we focus on the similarity-based method BSNet [27] and uncover why BSNet outperforms DIDS. We depict the t-SNE visualizations of their respective learned representations. As shown in Fig. 4(a) (b), we observe that the representations of BSNet in different attack types are more separated in the latent space. It can be inferred as one of the main causes of higher F1 scores in BSNet. Following this thread, we further depict their representation correlations via correlation heatmaps. We have found that the representations of BSNet are more entangled than DIDS as shown in Fig. 4(c) (d). Aligning with the disentanglement studies in Challenge 1, we attribute the main obstacle to the performance improvement of BSNet to the entangled representations. More discussions on the above empirical studies are available in Appendix 18.3.

In light of the discussions, we raise two critical questions:

- *Q1: How can an intrusion detection model automatically address the first challenges, i.e., two entangled distributions, to benefit the detection of both unknown and known attacks*
- *Q2: Can a few-shot intrusion detection method automatically learn the representations as separated in latent space, simultaneously as disentangled among representation elements?*

Achieving $Q1$ is challenging. To mitigate the issue of the first entangled distribution, we need to differentiate tens and thousands of features involved in real-time network traffic, without prior knowledge of the statistical distributions. Such a problem is largely under-explored in the field of NIDS. For the second entangled distribution, there are some remotely related methods [28]–[30] in other fields, including computer vision and natural language processing. However, these approaches

mainly focus on object-level representation learning, and hence they are hardly directly applied to intrusion detection to tackle this challenge.

Second, the difficulties of $Q2$ lie in two aspects: 1. tackling $Q2$ involves two optimization terms, a disentanglement term, and a latent space term. How to optimize them into an end-to-end framework is under-explored in NIDS and few-shot learning. 2. As for latent space optimization, existing similarity-based methods [27], [31], [32] mainly focused on utilizing the original-scale information of representations, while ignoring their multi-scale information to help few-shot traffic separated in latent space. We also provide formal definitions and illustrative figures for each case of entanglement and disentanglement in Appendix 11.

To address the above two questions, we propose a novel model called **D**isentangled dynamic **I**ntrusion **D**etection **S**ystem with **M**ulti-scale **F**ew-shot **L**earning (DIDS-MFL). Specifically, DIDS-MFL has two critical components: DIDS and MFL. The former DIDS disentangles the statistical flow features with a non-parametric optimization, aiming to automatically separate entangled distributions for representation learning. We refer to this step as statistical disentanglement. Then DIDS further learns to differentiate the representations by a regularization function, aiming to highlight the salient features for specific attacks with smaller coefficients. We refer to this step as representational disentanglement. DIDS finally introduces a novel graph diffusion module that dynamically fuses the graph topology in evolving traffic. The latter MFL is a flexible plug-and-play module for few-shot detection matching multiple NIDS baselines. We first optimize the multi-scale traffic representations in latent space to distinguish attack-specific information. Then we propose an element-wise disentanglement term to highlight the attack-specific information. Finally, we alternately solve them in an end-to-end framework.

Extensive experiments on five benchmarks show the superiority of our DIDS-MFL. The main contributions are:

- We propose DIDS-MFL, aiming to mitigate the entangled distributions of flow features for NIDS in various practical scenarios, e.g., known, few-shot, and unknown attacks [2]. To the best of our knowledge, we are the first to quantitatively analyze such an interesting problem and empirically reveal the underlying cause in the intrusion detection field.
- As for DIDS, we present a double disentanglement scheme for differentiating the general features of various attacks and highlighting the attack-specific features, respectively. We additionally introduce a novel graph diffusion method for dynamic feature aggregation.
- As for MFL, we propose a fusion-based framework by capturing multi-scale information and disentangling representations. We alternately solve them with rigorous derivation for few-shot intrusion detection. As a plug-and-play module, MFL also shows versatility in multiple baselines and downstream tasks.

---

[2]More details on unknown attack detection are available in Appendix 17.1.

- Extensive quantitative and qualitative experiments on various benchmarks demonstrate the effectiveness of DIDS-MFL and provide some helpful insights for effective NIDS in various practical scenarios.

II. RELATED WORK

A. Network Intrusion Detection System

Existing NIDS can be classified into two groups, i.e., signature-based ones [9]–[11] and anomaly-based ones [12]–[14], [33]. Signature-based intrusion detection systems (IDS) rely on identifying known attack patterns or signatures. These systems are adept at recognizing established threats, where the attack signatures are predefined, employing rule sets to match against network traffic for indicative malicious activities. However, their dependency on signature databases poses a limitation, as they are typically ineffective against novel or zero-day attacks that do not yet have a defined signature. In contrast, the anomaly-based ones do not rely on prior knowledge of attack signatures. They leverage machine learning or statistical techniques to model normal network behavior and flag deviations as potential threats. Anomaly detection holds the advantage of identifying unknown attacks and adapting to new threat patterns. The anomaly-based ones involve statistical methods [2], [3] and deep learning-based ones [15], [16]. Early deep learning studies model the traffic as independent sequences [34]–[37]. Recent popular studies rely on GCN to aggregate traffic information [38]–[40]. Most related to our work is Euler [41], which builds a series of static graphs based on traffic flow and then performs information aggregation. Our DIDS differs from the above methods in two aspects: 1) We build dynamic graphs rather than static ones, and such a dynamic aggregation can capture fine-grained traffic features for attack detection. Furthermore, we fuse the network layer information into our graph aggregation. 2) We introduce a double disentanglement scheme, including statistical disentanglement and representational one, to benefit the detection of both known and unknown attacks.

B. Disentangled Representation Learning

Disentanglement aims to learn representations that separate the underlying explanatory factors responsible for variation in the data. Previous studies [42]–[45] focus on the generative models by employing constraints on the loss functions, such as β-VAE [46] modifying the VAE framework to emphasize the independence of latent variables. Beta-VAE achieves disentanglement by introducing a trade-off term between reconstruction and KL divergence, while FactorVAE [47] introduces a total correlation penalty. CausalVAE [48] takes a significant step by integrating causal reasoning, allowing for interventions in the generative process. Disentangled Graph Neural Networks are another active area, where methods aim to learn independent node representations decoupled from the graph structure, e.g., GD-GAN employs a GAN framework to disentangle the latent factors of variation in graph data, allowing for controllable generation of new graph structures and node features. Some recent approaches [49]–[52] capture the intrinsic factors in graph-structured data. Most related to our work is DisenLink [52], which disentangled the original features into a fixed number of factors, with selective factor-wise message passing for better node representations. While our DIDS uses a double disentanglement method, which first disentangles the statistical features via non-parametric optimization, and then learns to highlight the attack-specific features with a regularization.

C. Dynamic Graph Convolution Networks

Dynamic graph convolution networks (GCNs) focus on evolving graph streams. There is a line of early studies in GCNs on dynamic graphs, which incorporate temporal information into graphs. These methods can be categorized into the spatio-temporal decoupled ones [39], [53], [54] and the spatio-temporal coupled ones [40], [55]–[57]. The former employs two separate modules to capture temporal and spatial information, e.g., DynGCN [58] performs spatial and temporal convolutions interleaved, updating model parameters to adapt to new graph snapshots, DIDA [59] Handles spatio-temporal distribution shifts by discovering invariant patterns and using intervention mechanisms to eliminate spurious impacts. The latter incorporates spatial-temporal dependencies by proposing a synchronous modeling mechanism, e.g., AST-GCN [60] adapts the graph structure and convolutional filters based on the temporal context to capture the evolving relationships in dynamic graphs, TD-GCN [61] uses temporal difference learning to update spatial-temporal graph convolutional filters, capturing the changes in graph structure and node states over time. Our DIDS is mainly inspired by the GIND [56], which adaptively aggregates information via a non-linear diffusion method. The key difference between our GCN approach and GIND is: we introduce the non-linear graph diffusion method into a multi-layer graph that considers the network topology for dynamic intrusion detection.

D. Few-shot Learning

Few-shot learning is a machine learning paradigm designed to enable a model to learn from a small number of training instances and generalize effectively to unseen data. The existing few-shot learning methods can be divided into two categories: meta-learning-based ones [62]–[64], including meta-learning-based NIDS approaches [65]–[68], and similarity-based ones. Meta-learning-based methods focus on learning a model that can adapt to new tasks with limited data. These methods often involve a meta-training phase where the model learns to learn from a variety of tasks, followed by a meta-testing phase where the model rapidly tunes to a new task. In contrast, the similarity-based ones rely on measuring the similarity or distance between the query instances and the support set to make predictions without necessitating vast types of few-shot samples. The similarity-based ones can also be grouped into augmentation-based ones [69]–[73], metric learning-based ones [32], [74]–[77], and other similarity-based few-shot learning methods [78]–[81]. Recent popular few-shot learning methods mainly focused on utilizing multiple learning metrics, or prototype completion. The most related to us is BSNet [27]. Ours differs from it in two aspects: first, we

emphasize the importance of preserving multi-scale intrinsic representation information, rather than optimizing the original-scale information; second, we propose a joint optimization scheme to generate separated and disentangled representations to make few-shot threats distinguishable.

Finally, this work is an extension version form [20]. In [20], we address the performance inconsistency issue caused by entangled features on sufficient training data. More practical scenarios is collecting attack traffic with limited instances. It is more challenging compared to [20] with involving double optimization goals. To this end, this paper proposes an alternating optimization framework MFL to address the entangled representations in few-shot traffic threats with rigorous derivation. The proposed MFL is also a plug-and-play few-shot intrusion detection module that is compatible with other NIDS methods.

III. PRELIMINARIES

A. Multi-Layer Graphs

To model the sophisticated traffic network topology, we formally define the multi-layer graphs. First, we consider a single-layer network modeled by a graph $G = (V, E, \omega)$, where V is the set of nodes and $E \subset V \times V$ is the set of edges. Here $\omega : V \times V \mapsto R$ is an edge weight function such that each edge $e_{uv} \in E$ has a weight ω_{uv}. Then the multi-layer graph can be defined as follows:

$$\mathbb{A} = \begin{pmatrix} A_{(1,1)} & \cdots & A_{(1,k)} & \cdots & A_{(1,m)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{(l,1)} & \cdots & A_{(k,k)} & \cdots & A_{(l,m)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{(m,1)} & \cdots & A_{(m,k)} & \cdots & A_{(m,m)} \end{pmatrix}. \quad (1)$$

where $A_{i,i}$ refers to the intra-layer adjacency matrix and $A_{i,j}(i \neq j)$ refers to cross-layer adjacency matrix.

B. Edge Construction

We have constructed a multi-layer dynamic graph by defining the devices as nodes and the communications between two devices as edges. To construct the edge relationships in the multi-layer graph, we first transform the Netflow data to edges by the following definition:

$$\mathbf{E}_{ij}(t) = (v_i, l_i, v_j, l_j, t, \Delta t, \mathbf{F}_{ij}(t)). \quad (2)$$

First, we concatenate the source IP and source port in the original traffic flows as the source identity for the device i. Similarly, we can obtain the destination identity for the device j. We denote v_i and v_j as the source and destination nodes respectively. Secondly, l_i denotes the layer of device i, and $l_i = 0$ indicates that i is a terminal device such as a PC, a server, or an IoT device. Here $l_i = 1$ indicates that i is an intermediate device such as a router in the communication link. Specifically, we assign devices with the router address 192.168.0.1 or with many stable connections in layer 1. Then t refers to the timestamp of traffic and Δt indicates the traffic duration time. Finally, $\mathbf{F}_{ij}(t)$ is the traffic features.

C. Problem Formulation

1) Dynamic Traffic Intrusion Detection: We first define the devices as nodes and the communications with timestamps between any pair of devices as edges. We use T to represent the maximum timestamp. An Edge sequence $\mathbb{E}$ is denoted by $\{\mathcal{E}^t\}_{t=1}^T$, where each $\mathcal{E}^t$ represents a network traffic. Also, after each edge, there is a corresponding multi-layer graph, then the corresponding multi-layer graph stream $\mathbb{G}$ takes the form of $\{\mathcal{G}^t\}_{t=1}^T$, where each $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ represents the multi-layer graph at timestamp t. A multi-layer adjacency matrix $\mathbb{A}^t \in \mathbb{R}^{m \times n}$ represents the edges in $\mathcal{E}^t$, where $\forall (i, j, w) \in \mathcal{E}^t, \mathbb{A}^t[i][j] = w_{ij}$ and w_{ij} is the weight of the matrix. The goal of intrusion detection is to learn to predict the edge $\mathcal{E}^t$ as a benign traffic or an attack in binary classification, and a specific type under the multi-classification.

2) Few-shot Intrusion Detection: Given a few-shot intrusion detection task $T = \{C_{\text{tr}}, C_{\text{te}}\}$, where C_{tr} is a training set with few-shot traffic, and C_{te} is a test set. Our goal is to design a few-shot module to accurately classify the samples in C_{te} by few-shot learning on C_{tr}. C_{tr} includes two key components, support set and query set. The former provides the few-shot traffic information to the model, and the latter contains new instances for model evaluation and optimization. The query set can be sampled from the support set for the traffic-limited scenarios. Specifically, we conduct $\mathcal{N}$-way $\mathcal{K}$-shot intrusion detection, $\mathcal{N}$-class traffic with $\mathcal{K}$ samples, i.e., 5, in the support set. The few-shot intrusion detection model aims to conduct accurate detection in the query set, while the only available reference is the few-shot traffic in the support set.

IV. MODEL

In this section, we present the DIDS-MFL, which consists of five main modules. Figure 5 shows the architecture of our proposed model. Next, we dive into the details of these modules.

A. Statistical Disentanglement

As we discussed in Section I, statistical distributions of traffic features are one of the main underlying causes of performance variations. i.e., separated distributions benefit the unknown attack identification, while entangled ones are indistinguishable and thus unable to help the NIDS to make accurate decisions. Therefore, our aim is there to disentangle the traffic features and make them distinguishable.

To separate the features of traffic without any prior knowledge, we formulate the differentiation as a constrained nonparametric optimization problem and approximate the optimal results by solving the Satisfiability Modulo Theory (SMT) [82]. We perform a min-max normalization on the edge feature $\mathbf{F}_{ij}(t)$. For convenience, we denote the normalized edge feature as $\mathcal{F}$, and $\mathcal{F}_i$ is the i-th normalized element.

We need a weight matrix $\mathbf{w}$ to generate the disentangled representation of $\mathcal{F}$. Our key optimization objectives are to minimize the mutual information between the elements of traffic features and also bound the range of $\mathbf{w}$ when we perform aggregation. We start by constraining the weight
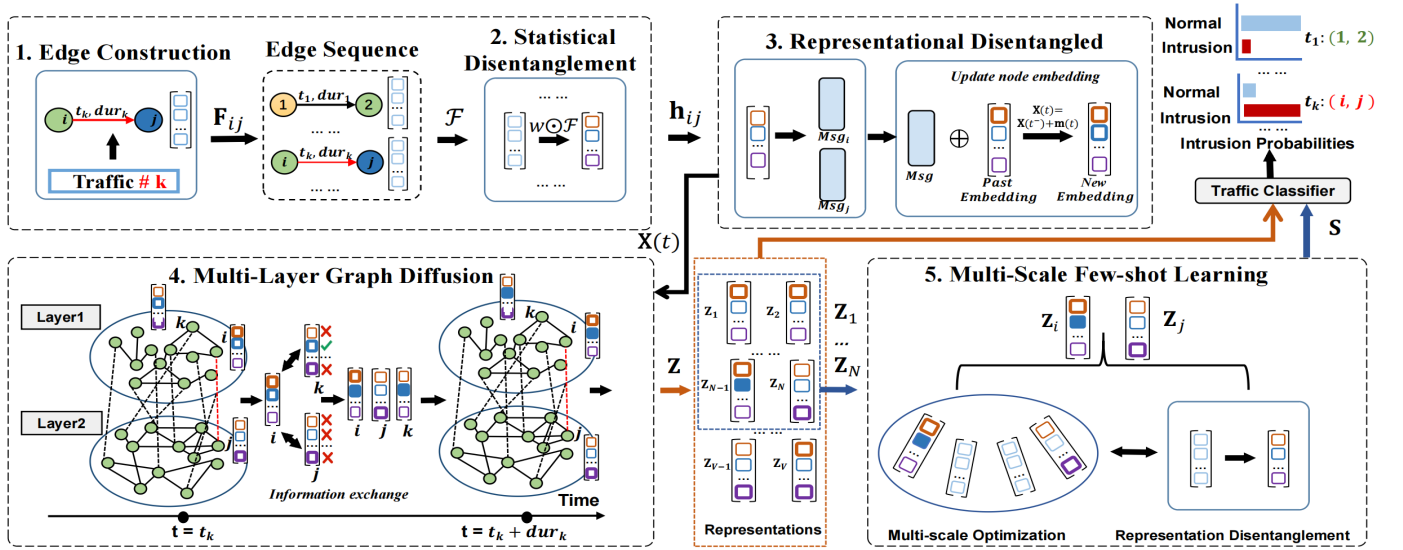
Fig. 5. Overview of the proposed DIDS-MFL, which consists of five modules. 1) Edge construction module builds edges based on traffic flow. 2) Statistical disentanglement module differentiates values in vectors to facilitate the identification of various attacks. 3) Representational disentanglement module learns to highlight attack-specific features. 4) Multi-Layer graph diffusion module fuses the network topology for better aggregation over evolving dynamic traffic. 5) Multi-scale few-shot learning module aims to few-shot traffic threats detection. Finally, traffic classifier takes the traffic representation as an input to yield the detection results. DIDS takes the first four steps and flows through the orange arrows to the classifier, while DIDS-MFL takes the five steps by flowing through the blue arrows to the classifier.

matrix $\mathbf{w}$ with the range of the superposition function as follows:

$$W_{\min} \leq \mathbf{w}_i \leq W_{\max} \quad (1 \leq i \leq N), \quad \sum_{i=1}^{N} \mathbf{w}_i \mathcal{F}_i \leq B, \quad (3)$$

where $W_{\min}$, $W_{\max}$ and B are constants, N is the edge feature dimension. We then constrain the order-preserving properties of generated representations:

$$\mathbf{w}_i \mathcal{F}_i \leq \mathbf{w}_{i+1} \mathcal{F}_{i+1} \quad (1 \leq i \leq N-1). \quad (4)$$

Finally, we maximize the distance between components in the vector $\mathbf{w}$, consequently minimizing the mutual information between each two feature elements. In this way, we can disentangle the distribution of element-wised features. The optimization objective can be expressed as follows:

$$\widetilde{\mathbf{w}} = \arg\max(\mathbf{w}_N \mathcal{F}_N - \mathbf{w}_1 \mathcal{F}_1 - \sum_{i=2}^{N-1} |2\mathbf{w}_i \mathcal{F}_i - \mathbf{w}_{i+1}\mathcal{F}_{i+1} - \mathbf{w}_{i-1}\mathcal{F}_{i-1}|). \quad (5)$$

We are unable to determine the convexity of the optimization object due to its closed form. Therefore, we transform the above problem into an SMT problem with an optimization objective (6) and a subjection (7) to approximate the optimal results.

$$\widetilde{\mathbf{w}} = \arg\max(\mathbf{w}_N \mathcal{F}_N - \mathbf{w}_1 \mathcal{F}_1 + \sum_{i=2}^{N-1} 2\mathbf{w}_i \mathcal{F}_i - \mathbf{w}_{i-1}\mathcal{F}_{i-1} - \mathbf{w}_{i+1}\mathcal{F}_{i+1}), \quad (6)$$

subjects to:

$$\begin{cases} \mathbf{w}_i & \in & [W_{\min}, W_{\max}] \\ \sum_{i=1}^{N} \mathbf{w}_i \mathbf{n}_i & \leq & B \\ \mathbf{w}_i \mathcal{F}_i & \leq & \mathbf{w}_{i+1} \mathcal{F}_{(i+1)} \\ 2\mathbf{w}_i \mathcal{F}_i & \leq & \mathbf{w}_{i-1}\mathcal{F}_{i-1} + \mathbf{w}_{i+1}\mathcal{F}_{i+1}. \end{cases} \quad (7)$$

The subjection in Eq. 7 ensures variants' ranges and the order-preserving property of generated representation. It also ensures symbols when removing the absolute value sign. The detailed explanation of why Eq. 6 can generate the disentangled representation is available in Appendix 7.

Specifically, we employ Eq. 6 as an optimization objective, i.e., loss function, and utilize Adam optimizer to solve it. Then we can generate the disentangled edge representation $\mathbf{h}_{i,j}$, which can be expressed as $\mathbf{h}_{i,j} = \mathbf{w} \odot \mathcal{F}$, where the symbol $\odot$ represents the Hadamard product.

Equipped with the above non-parametric optimization, we can differentiate tens and hundreds of complex features of various attacks, mitigating the entangled distribution of statistical features. Such statistically disentangled features facilitate our model to be more sensitive to various attacks.

B. Representational Disentanglement

So far we have constructed edges and statistically differentiated the features of traffic flows $\mathbf{h}_{i,j} = \mathbf{w} \odot \mathcal{F}$. This module generates contextualized **node representations $\mathbf{X}$** from edge representations $\mathbf{h}_{i,j}$ by using the temporal information. This involves three steps:

1) Generating updating messages: For an incoming traffic flow, we will build an edge or update the corresponding edge, which may lead to a dramatic change in the node representations involved in this interaction. We can update the node representations by utilizing this change. Therefore,

we name the abrupt change an updating message and denote it as $\mathbf{c}(t)$. Specifically, we generate $\mathbf{c}(t)$ by incorporating historical memory and disentangled edge representation $\mathbf{h}_{i,j}$. The messages of node i and j are as follows:

$$\mathbf{c}_i(t) = \mathrm{Msg}\left(\mathbf{m}_i\left(t^-\right), \mathbf{m}_j\left(t^-\right), t, \Delta t, l_i, l_j, \mathbf{h_{i,j}}\right), \quad (8)$$

$$\mathbf{c}_j(t) = \mathrm{Msg}\left(\mathbf{m}_j\left(t^-\right), \mathbf{m}_i\left(t^-\right), t, \Delta t, l_i, l_j, \mathbf{h_{i,j}}\right), \quad (9)$$

where $\mathbf{h}_{i,j}$ is the disentangled edge representation. Δt is the edge duration time, l_i, l_j is the layer marks of edge, $\mathbf{m}_i(t^-)$ is the historical memory of the two interacting nodes, where t^- is a historical time point, compared to the existing time point t, Msg is a learnable function, and we use RNN. The initial memory $\mathbf{m}_i(0)$ is $\mathbf{0}$, $\forall i \in V$. Specifically, RNN aims to generate the new memory message $\mathbf{c}(t)$ combining the historical memory $\mathbf{m}(t^-)$ and disentangled representation $\mathbf{h}_{ij}$ by learnable weight matrices and gating mechanism.

2) Updating node memory: We update the node memory by merging the latest message with the historical memory and then encode the merging information, which can be expressed as follows:

$$\mathbf{m}_i(t) = \mathrm{Mem}\left(\mathbf{c}_i(t), \mathbf{m}_i\left(t^-\right)\right), \quad (10)$$

where Mem is an encoder, and here we use GRU. We similarly update the memory of node j via (10). Specifically, GRU introduces Reset Gate and Update Gate to better generate the updated node memory $\mathbf{m}_i(t)$ by fusing historical memory $\mathbf{m}_i(t^-)$ and new memory message $\mathbf{c}_i(t)$. GRU considers both long-term and short-term dependencies, thus mitigating the vanishing gradient issue in temporal models.

3) Generating second disentangled node representations: We can generate the node representation by utilizing the updated memory in (10), and the representation of node i in time t can be expressed as $\mathbf{x}_i(t) = \mathbf{x}_i(t^-) + \mathbf{m}_i(t)$, where $\mathbf{x}_i(t^-)$ is the historical representation of node i. In this way, we obtain the dynamic node representations of traffic in an evolving time flow. The representation of node j can also be generated in a similar way. The initial value of node representations $\mathbf{X}(0)$ is $\mathbf{0}$.

We aim to preserve the disentangled property in node representations. However, the update operations above may entangle them at the element level again. Therefore, we propose the second representational disentanglement, which aims to highlight the attack-specific features in node representations in the following end-to-end manner. It also ensures that the element-wised representations are close to orthogonal.

$$\mathcal{L}_{\mathrm{Dis}} = \frac{1}{2}\left\|\mathbf{X}(t)\mathbf{X}(t^-)^\top - \mathbf{I}\right\|_F^2. \quad (11)$$

The above regularization encourages the model to learn smaller coefficients between every two elements of node representations. Such representations can be more differentiated. As the module is supervised by the signal of a specific attack, it can learn to highlight the attack-specific features, so as to improve the detection accuracy, as discussed in Section I. By doing so, we are able to mitigate the entangled distribution of representational features as mentioned at the beginning.

C. Multi-Layer Graph Diffusion

So far, we have generated the dynamic disentangled node representations with temporal information. For further fusing the multi-layer topological structure information, we propose a multi-layer graph diffusion module. Please note that we still preserve the disentangled property by the following customized designs.

We utilize the following graph diffusion method to fuse the topological information in evolving graph streams, which can capture the fine-grained spatial-temporal coupled information. The previous dynamic intrusion detection methods may lose this information in separated time gaps, due to the employed time-window or snapshots-based methods. More discussions on the 'dynamic' and superiority of the proposed graph diffusion approach compared to the previous methods, are available in Appendix 17.2.

$$\begin{cases} \partial_t \mathbf{X} = \mathbf{F}(\mathbf{X}, \mathbf{\Theta}) \\ \mathbf{X}(0) = \mathbf{0}, \end{cases} \quad (12)$$

where $\mathbf{F}$ is a matrix–valued nonlinear function conditioned on graph $\mathcal{G}$, and $\mathbf{\Theta}$ is the tensor of trainable parameters. The above Eq. 12 establishes the foundation for spatial-temporal coupled information modeling.

Specifically, we aim to amplify the important dimensions of disentangled representations and depress the influence of trivial feature elements in the diffusion process. To formulate this process, we consider PM (PeronaMalik) diffusion [83], a type of nonlinear filtering. It can be expressed as:

$$\begin{cases} \frac{\partial x(u,t)}{\partial t} = \mathrm{div}[g(|\nabla x(u,t)|)\nabla x(u,t)] \\ x(u,0) = 0, \end{cases} \quad (13)$$

where div is the divergence operator, ∇ is the gradient operator, and g is a function inversely proportional to the absolute value of the gradient.

Before formally formulating the multi-layer graph diffusion, we propose the following spatial-temporal influence coefficient $s_{ij} \in \mathbf{S}$ between nodes i,j in t_{ij} time. The coefficient matrix considers the information changes over spatial-temporal coupled traffic data. Specifically, different layers and topology structures over traffic networks and the traffic feature interactions at different times will influence the node representations dynamically:

$$s_{ij} = f(l_i||l_j||\phi(t - t_{ij})), \quad (14)$$

$$f(\mathbf{x}) = \mathbf{W}^{(2)} \cdot \mathrm{ReLU}\left(\mathbf{W}^{(1)}\mathbf{x}\right), \quad (15)$$

where $||$ is a concatenate operator, $\phi(\cdot)$ is a generic time encoder [84] to generate temporal representations, $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are the parameters of the first and second layer MLP.

Now we formally propose the multi-layer graph diffusion module. We first define a differential operator on the multi-layer graph, aiming to transfer the above continuous PM diffusion in Eq. 13 to multi-layer graphs. As known from previous literature [85], the gradient operator corresponds to the instance matrix $\mathbf{M}$, while the divergence operator corresponds to the matrix $\mathbf{M}^\top$, and we can compute the matrix $\mathbf{M}$ by the equation $\mathbf{M}^T\mathbf{M} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal

matrix. Then our novel multi-layer diffusion can be expressed as:

$$\partial \mathbf{X}_t = -\mathbf{M}^\top \sigma(\mathbf{MXK}^\top)\mathbf{S}\left(\mathbf{MXK}^\top\right)\mathbf{K}, \quad (16)$$

where $\mathbf{K}$ is a transformation matrix, $\mathbf{S}$ is the structure-temporal influence coefficients calculated in Eq. 14 and $\sigma(\cdot)$ represents the function $\exp(-|\cdot|)$. The solution to equation 16 can be expressed as:

$$\mathbf{X}_{t+\Delta t} = \mathbf{X}_t + \int_t^{t+\Delta t} \partial_t \mathbf{X}_t d\tau, \quad (17)$$

where t is the last edge occurrence time and Δt is the edge duration and we use the Runge-Kutta method to solve this equation.

D. Classifier and Loss Function

For DIDS, we make the two-step predictions for intrusion detection. We utilize the first MLP to classify whether the traffic is benign or anomalous and utilize the second MLP to detect the specific type of attack. When an unknown attack invades, the direct multi-classifications will be easy to fail to assign the anomalous label thus leading to poor performance. In contrast, through the first-step binary classification, DIDS will focus more on inconsistencies with normal behavior to improve the performance of detecting unknown attacks. The following second multi-classification will further alert the administrators that what kind of attack it is more similar to so that similar mitigation measures can be taken. Specifically, the intrusion loss can be expressed as:

$$\mathcal{L}_{\text{Int}} = -\sum_{i=1}^m (\log(1 - p_{\text{nor},i}) + \log(p_{\text{att},i}) + \sum_{j=1}^K y_{i,k}\log(p_{i,k})) \quad (18)$$

where m is the batch size, K is the number of attack classes, $p_{\text{nor},i}$ is the probability of normal, $p_{\text{att},i}$ is the probability of attack. Additionally, the adjacent time intervals may cause adjacent times embedding to be farther apart in embedded space, due to the learning process independency. To address this problem, we constrained the variation between adjacent timestamps embedding by minimizing the Euclidean Distance:

$$\mathcal{L}_{\text{Smooth}} = \sum_{t=0}^T \|\mathbf{X}_{t+\Delta t} - \mathbf{X}_t\|_2. \quad (19)$$

Finally, the overall loss of DIDS can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{\text{Int}} + \delta\mathcal{L}_{\text{Smooth}} + \zeta\mathcal{L}_{\text{Dis}}, \quad (20)$$

where δ, and ζ are trade-off parameters.

E. Multi-scale Few-shot Learning

So far we have proposed DIDS for supervised intrusion detection. Furthermore, we propose the following MFL to detect few-shot threats accurately, e.g., 5 samples of each attack type. We first denote the learned representations $\mathbf{X}_{t+\Delta t}$ in Eq. 17 as $\mathbf{Z} \in \mathcal{R}^{L*N}$, where L is the length of representations, and N is the few-shot sample number. The few-shot learner aims to generate a coefficient matrix $\mathbf{S}$ as a learned similarity matrix

between each sample pair, thus benefiting the distinction of few-shot traffic threats.

Specifically, we propose a Multi-scale Few-shot learning (MFL) module. MFL first focuses on the original representations $\mathbf{Z}$ to generate a coefficient matrix as:

$$\min_{\mathbf{H}} = \|\mathbf{Z} - \mathbf{ZH}\|_F^2 + \lambda_1\|\mathbf{H}\|_F^2, s.t.\text{diag}(\mathbf{H}) = 0, \quad (21)$$

where $|\cdot|_F^2$ is the Frobenius matrix norm (F-norm) and $\text{diag}(\mathbf{H})$ denotes the diagonal entries of $\mathbf{H}$. $\mathbf{H}$ can be directly derived by solving Eq. 21. Specifically, we directly employ Eq. 21 as a loss function and utilize backpropagation and optimizer, e.g., Adam, to solve it.

The above optimization term in MFL only preserves the original-scale information of $\mathbf{Z}$. The multi-scale information of $\mathbf{Z}$ can also provide necessary information to make few-shot traffic samples distinguishable. To this end, we proposed a transform-based algorithm to generate coefficient matrix $\mathbf{Q}$ by using $\mathbf{Z}$'s multi-scale information. This process can be considered as an 'augmentation' of the original representations $\mathbf{Z}$ in the latent space, aiming to capture the invariant attack-specific features across different scales. Finally, we fuse the obtained coefficient matrices $\mathbf{H}$ and $\mathbf{Q}$. Equipped with the fusion of generated matrices, MFL can effectively utilize few-shot representations across different scales, i.e., under the original and transformed scale, thus benefiting the distinction of few-shot traffic threats. We have the following principles to achieve MFL and clarify the motivations behind these designs:

- 1. MFL should encourage the representations under different scales to be close in latent space, thus capturing the attack-specific features and benefiting the distinction of traffic samples in different types, which is motivated by the empirical studies in Fig. 4 (a)(b)
- 2. MFL should ensure the disentanglement among the elements of the learned representation, thus highlighting the attack-specific information, which is motivated by the empirical studies in Fig. 4 (c)(d)

To achieve the above goals, we first propose a transformation-based learning scheme by projecting the representations $\mathbf{Z}$ to a latent space, aiming to discover the attack-specific invariant features across different scales. Then we introduce a regularization term to disentangle the few-shot representations. Finally, we propose an alternating optimization algorithm to derive $\mathbf{Q}$ in an end-to-end way. The implementation involves the following three steps:

1) Generating Representation Transformation.

We denote $\mathbf{Z}_o = \mathbf{Z}$ as the original representations, and $\mathbf{Z}_t = \mathcal{G}(\mathbf{Z})$ as the transformed representations, where $\mathcal{G}$ is a transformation operator. Specifically, we introduce a scaling operator $\mathcal{S}$ as follows:

$$\mathcal{S} = \text{diag}(s_1, s_2, ..., s_L) = \begin{bmatrix} s_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_L \end{bmatrix}, \quad (22)$$

where $s_1, s_2, ..., s_D$ is the scale factor, and different values determine different scalars. Then the scaling transformation can be written as $\mathbf{Z}_t = \mathcal{G}(\mathbf{Z}) = \mathcal{S}\mathbf{Z}$. Especially, when $s_1 = s_2 = ... = s_L = \gamma$, $\mathcal{S}$ is an equal-rate scaling operator with the

rate of γ. Hence, the transformation can be rewritten as $\mathbf{Z}_t = \mathcal{G}(\mathbf{Z}) = \mathcal{S}\mathbf{Z} = \gamma\mathbf{I}\mathbf{Z} = \gamma\mathbf{Z}$. By the equal-rate transformation, we generate transformed representations while preserving their content, e.g., scaling up an image by the rate of 2, will not change the content we see.

2) Generating Multi-scale Coefficient Matrix.

So far, we have generated the transformations of $\mathbf{Z}$. In step 2, we propose a constraint term to encourage the representation to be close between the original and transformed scale. Furthermore, we introduce a learnable projection operator $\mathbf{P}$ to uncover the attack-specific invariant information, thus benefiting the distinction of few-shot traffic threats.

$$f_1(\mathbf{P}) = \left\| \mathbf{P}^T\mathbf{Z}_o - \mathbf{P}^T\mathbf{Z}_t \right\|_F^2, \tag{23}$$

The projection operator $\mathbf{P}$ can be considered as a metric matrix across different-scale representation, also benefiting the connectivity of same-attack traffic samples under the transformation. Then we formally propose an optimization problem as follows:

$$\begin{aligned} \min\ &f_1(\mathbf{P}) + \eta\|\mathbf{Q}\|_F, \\ \text{s.t. } &\mathbf{P}\mathbf{Z} = \mathbf{P}\mathcal{G}(\mathbf{Z})\mathbf{Q}, \end{aligned} \tag{24}$$

The above formulation Eq. 24 corresponds to design principle 1 and is motivated by the empirical studies in Fig. 4 (a)(b), aiming to distinguish the attack-specific information in the latent space.

3) Disentangling the Learned Representations.

Finally, we introduce a disentangle regularization term to highlight the attack-specific information, thus mitigating the representation entangled problem in few-shot traffic samples.

$$f_2(\mathbf{P}) = -\sum_j \left\| \mathbf{P}^T\mathbf{Z}_o^{+j} - \mathbf{P}^T\mathbf{Z}_t^{-j} \right\|_F^2, \tag{25}$$

where $\mathbf{Z}_o^{+j} \in \mathbb{R}^{(L-1)\times N}$ repeats the j-th row of $\mathbf{Z}_o$ and $\mathbf{Z}_t^{-j} \in \mathbb{R}^{(L-1)\times N}$ denotes the transformation matrix by removing the j-th row. The above disentanglement regularization term Eq. 25 corresponds to design principle 2 and is motivated by the empirical studies in Fig. 4 (c)(d), aiming to highlight the attack-specific information.

Combining the above key formulations Eq. 24 and Eq. 25, we can rewrite the optimization problem as:

$$\begin{aligned} \min\ &f(\mathbf{P}) + \eta\|\tilde{\mathbf{Q}}\|_F, \\ \text{s.t. } &\mathbf{P}\mathbf{Z} = \gamma\mathbf{P}\mathbf{Z}\mathbf{Q}, \tilde{\mathbf{Q}} = \mathbf{Q}, \end{aligned} \tag{26}$$

where $f(\mathbf{P}) = \frac{\alpha}{2}(f_1(\mathbf{P}) + f_2(\mathbf{P}))$, α and η are trade-off hyperparameters, $\tilde{Q}$ is a auxiliary variable, and γ is scaling rate hyperparameter.

Since the optimization problem in Eq. 26 is not convex with the unknowns $\{\mathbf{P}, \mathbf{Q}\}$, we solve $\mathbf{Q}$ by iteratively updating variables while fixing another. We propose the following alternating solution to derive the coefficient matrix $\mathbf{Q}$.

Alternating Solution for MFL

We solve Eq. 26 by converting the original problem to the augmented Lagrange minimizing problem, as Eq. 26 involving a multi-objective optimization:

$$\begin{aligned} \mathcal{L} = &\frac{\alpha}{2} \left\| \mathbf{P}\mathbf{Z} - \gamma\mathbf{P}\mathbf{Z}\mathbf{Q} \right\|_F^2 \\ &- \frac{\alpha}{2} \sum_j \left\| \mathbf{P}\mathbf{Z}^{+j} - \gamma\mathbf{P}\mathbf{Z}^{-j} \right\|_F^2 + \eta\|\tilde{\mathbf{Q}}\|_F \\ &+ <\Phi_1, \mathbf{P}\mathbf{Z} - \gamma\mathbf{P}\mathbf{Z}\mathbf{Q} + <\Phi_2, \mathbf{Q} - \tilde{\mathbf{Q}}> \\ &+ \frac{\mu}{2} \left(\|\mathbf{P}\mathbf{Z} - \gamma\mathbf{P}\mathbf{Z}\mathbf{Q}\|_F^2 + \|\mathbf{Q} - \tilde{\mathbf{Q}}\|_F^2 \right), \end{aligned} \tag{27}$$

where Φ_1 and Φ_2 are the Lagrange multipliers and the penalty parameter $\mu > 0$.

The alternating solution for Eq. 27 involves four steps, and the details are available in Appendix 13.

So far, we have obtained coefficient matrices $\mathbf{H}$ and $\mathbf{Q}$ by parallel computing of Eq. 21 and Eq. 27 using the alternating solving algorithm. We can fuse two coefficient matrices by introducing a trade-off hyperparameter ϵ:

$$\mathbf{Q}^* = \mathbf{H} + \epsilon * \mathbf{Q} \tag{28}$$

where $\mathbf{Q}^*$ contains fine-grained multi-scale information to help few-shot traffic threats distinguishable.

Finally, we can generate the coefficient matrix of MFL by $\mathbf{S}^* = \left(|\mathbf{Q}^*| + |\mathbf{Q}^*|^T\right)/2$. Given the noise information and outliers in the obtained $\mathbf{S}^*$, in practical implementations, we further conduct SVD decomposition of $\mathbf{S}^*$ to filter noisy information and generate a normalized matrix $\mathbf{S}$. The detailed implementation pseudo-code of generating $\mathbf{S}$ is available in the Appendix 14.

DIDS-MFL Loss

The DIDS-MFL loss consists of two components, a multi-scale few-shot learning term and a regularization term. The former uses a cross-entropy loss, aiming to match the prediction to the query set via the generated multi-scale coefficient matrix $\mathbf{S}$. In the few-shot intrusion detection task, $\mathbf{S}$ is derived from the representations of support and query samples. The latter enforces the representations $\mathbf{Z}$ to be close to the original-scale representations in latent space. Finally, we introduce a trade-off hyperparameter β to control the regularization intensity. The specific DIDS-MFL loss is as follows:

$$\mathcal{L} = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{Q}} Y_{i,j} \log(\mathrm{P}_{i,j}) + \beta\|\mathbf{Z}\mathbf{H} - \mathbf{Z}\| \tag{29}$$

where $\mathcal{N}$ is the category number of attacks, $\mathcal{Q}$ is sample number of query set in each category, $Y_{i,j}$ and $\mathrm{P}_{i,j}$ are ground-truth and prediction, respectively, P_{ij} is query sample's mean similarity score derived from the coefficient matrix $\mathbf{S}$, β is a trade-off hyperparameter. In Eq. 29, the first term discovers the multi-scale information of $\mathbf{Z}$, while the second term preserves the information under the original scale via the coefficient matrix $\mathbf{H}$. We can fine-tune β to control these two terms.

V. EXPERIMENTS

A. Experimental Settings

Datasets: We conduct experiments on five popular datasets that involve massive network traffic over the internet of things (IoT). We give detailed descriptions as follows:

- CIC-ToN-IoT : This dataset is generated from the existing ToN-IoT dataset by a network traffic tool CICFlowMeter, where TON-IoT is a well-known database for intrusion detection collected from Telemetry datasets of IoT services. This dataset consists of $5,351,760$ flows, with 53.00% attack samples and 47.00% benign samples.
- CIC-BoT-IoT: It is generated from the existing BoT-IoT dataset by CICFlowMeter. This dataset consists of $6,714,300$ traffic flows, with 98.82% attack samples and 1.18% benign samples.
- EdgeIIoT: It is collected from an IoT/IIoT system that contains mobile devices and sensors. This dataset includes $1,692,555$ flows, with 21.15% attack samples and 78.85% benign samples.
- NF-UNSW-NB15-v2: It is NetFlow-based and generated from the UNSW-NB15 dataset, which has been expanded with additional NetFlow features and labeled with attack categories. This dataset includes $2,390,275$ flows, with 3.98% attack samples and 96.02% benign samples.
- NF-CSE-CIC-IDS2018-v2: It is a NetFlow-based dataset generated from the original pcap files of CSE-CIC-IDS2018 dataset. This dataset includes $18,893,708$ flows, with 11.95% attack samples and 88.05% benign samples.

Configurations: All experiments and timings are conducted on a machine with Intel Xeon Gold 6330@ 2.00GHz, RTX3090 GPU, and 24G memory. We use the Adam optimizer with a learning rate of 0.01, the learning rate scheduler reducing rate as 0.9, with weight decay being $1e^{-5}$. We train all the models with 500 epochs.

Baselines: To evaluate the performance of the proposed DIDS, we select 10 deep learning based-models as baselines, including 3 sequence models (i.e., MLP [86], MStream [16], LUCID [37]), 4 static GCN models (i.e., GAT [87] and E-GraphSAGE [88], SSDCM [89], DMGI [90]), where SSDCM and DMGI are designed for static multi-layer graphs, 4 dynamic GCN models (i.e., TGN [91], EULER [41], AnomRank [92], DynAnom [92]). Additionally, we choose 3 rule-based baselines to compare with the proposed DIDS, (i.e., ML [93], AdaBoost [94], and Logistic Regression).

Metrics: We follow the previous works [95] to evaluate the performances by two commonly used metrics in intrusion detection including F1-score (F1) and ROC-AUC score (AUC).

B. Main Results

1) Comparisons of binary classification: Under this setting, we classify a traffic flow as an attack or a benign one. We categorize the baselines into three groups, including dynamic GCNs at the top of Table I, static GCNs at the middle of the table, and another three baselines at the bottom. It should be noted that AnomRank and DynAnom are two popular baselines for anomaly detection. We run our experiment 5 times and report the mean and variance values. The comparison results in Table I show that our DIDS consistently performs the best among all baselines over the five benchmarks, which shows the superiority of our method for intrusion detection. Specifically, compared to the E-GraphSAGE, the previous state-of-art GCN-based approach, our method achieves a 4.80% higher F1-score over

the CIC-BoT-IoT dataset. Our DIDS outperforms AnomRank, the previous state-of-the-art method for anomaly detection on F1, by 15.27 points over the EdgeIIoT dataset. We attribute the above results to the gains of our statistical disentanglement, representational disentanglement and dynamic graph diffusion.

2) Comparisons of multi-classification: We compare the performance of our method to four baselines in declaring the specific attack type. These baselines include E-GraphSAGE [17], TGN [91], ML [98], and AdaBoost [99], as they are representative of different types of intrusion detection models and have been widely used in previous studies. Figure 6 demonstrates that the proposed DIDS consistently performs well among all five datasets, especially under some complex attacks. The classification results in Fig. 6 correspond to the second issue we claimed in Challenge 1. The existing graph-based methods, including E-GraphSAGE, ML, and AdaBoost, perform inconsistently in the identification of complex attacks (e.g., MITM and Backdoor). We also observe that some attacks that are not easily detected by the baseline approaches, can be identified by the proposed DIDS with higher F1 scores. For example, E-GraphSAGE only achieves 18.34% and 30.7% F1 scores on CIC-ToN-IoT for MITM and Backdoor attacks, respectively, while our DIDS is able to obtain higher than 40% F1 score gains for each attack. These results further show the superiority of the proposed DIDS. We also find that the average F1 scores of our method on CIC-BoT-loT and NF-CSE-CIC-IDS2018-v2 are lower than those on other datasets. The underlying reason is the unbalanced attack distributions in the training set, where the dominant type may mislead the classifications. Such a finding aligns with previous work in the field of computer vision [100]. Nevertheless, the proposed DIDS is still the best under such distributions. We leave this interesting observation as our future work.

3) Comparisons of unknown attacks: To further investigate the performance of detecting unknown attacks, we conduct experiments on the four attack types by discarding the corresponding instances in the train set and detecting them in the test set. From this perspective, the unknown attacks are the new-type attacks that do not belong to the existing types of attacks in the training set. The unknown attack detection results in Table III correspond to the first issue we claimed in Challenge 1. We run our experiments 5 times and report the mean and variance values with different random seeds. Table III reports the classification results on the CIC-ToN-IOT dataset. It shows that the statistical rule-based method Logistic Regression can only achieve as low as a 1.68 F1 score for DDoS attacks, this confirms our analysis at the very beginning that rule-based methods can hardly detect unknown attacks. The score of graph-based E-GraphSAGE is much smaller than DIDS, e.g., 6.05% for MITM, indicating the limitations of the static graph in detecting unknown attacks. We also observe that TGN performs better than E-GraphSAGE, although both of them are graph-based methods. We attribute the improvement to the dynamic module for TGN. Nevertheless, our DIDS outperforms all these methods by a large margin, with an average score of 33.65% on the four attacks. The results also suggest that our method is more consistent in detecting various unknown attacks, showing the effectiveness of the two

TABLE I
COMPARISONS OF BINARY CLASSIFICATION ON FIVE DATASETS. THE RESULTS WITH ‡ ARE DIRECTLY COPIED FROM [96].

Methods	CIC-TON-IoT		CIC-BoT-IoT		EdgeIIoT		NF-UNSW-NB15-v2		NF-CSE-CIC-IDS2018-v2	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
TGN [91]	$89.90_{\pm1.66}$	$82.09_{\pm1.36}$	$96.84_{\pm0.44}$	$94.41_{\pm0.81}$	$94.99_{\pm0.61}$	$89.50_{\pm2.04}$	$93.55_{\pm0.23}$	$88.01_{\pm1.97}$	$95.11_{\pm0.46}$	$91.30_{\pm0.76}$
EULER [41]	$89.73_{\pm1.13}$	$80.48_{\pm2.46}$	$96.00_{\pm0.29}$	$91.47_{\pm1.36}$	$92.89_{\pm0.32}$	$90.64_{\pm1.80}$	$92.76_{\pm0.86}$	$86.97_{\pm1.11}$	$95.87_{\pm0.51}$	$90.76_{\pm0.64}$
AnomRank [92]	$76.51_{\pm0.98}$	$77.41_{\pm1.64}$	$84.84_{\pm0.49}$	$82.50_{\pm0.59}$	$78.37_{\pm0.43}$	$81.36_{\pm0.41}$	$90.54_{\pm2.40}$	$79.63_{\pm0.12}$	$90.08_{\pm0.82}$	$83.76_{\pm0.35}$
DynAnom [97]	$79.23_{\pm1.81}$	$75.22_{\pm0.92}$	$83.25_{\pm0.62}$	$79.04_{\pm0.84}$	$81.56_{\pm0.94}$	$83.94_{\pm0.36}$	$89.11_{\pm1.48}$	$85.25_{\pm0.64}$	$91.21_{\pm0.95}$	$88.79_{\pm0.54}$
Anomal-E [96]	-	-	-	-	-	-	91.89‡	-	94.51‡	-
GAT [87]	$86.30_{\pm1.16}$	$74.66_{\pm1.37}$	$94.56_{\pm0.75}$	$93.09_{\pm2.83}$	$93.30_{\pm0.14}$	$88.30_{\pm1.56}$	$92.20_{\pm1.60}$	$89.91_{\pm0.62}$	$96.08_{\pm0.24}$	$90.56_{\pm0.34}$
E-GraphSAGE [17]	$89.46_{\pm1.25}$	$79.56_{\pm1.63}$	$93.74_{\pm0.76}$	$90.53_{\pm1.90}$	$92.10_{\pm1.46}$	$89.10_{\pm0.64}$	$94.10_{\pm0.33}$	$90.39_{\pm0.26}$	$95.71_{\pm0.35}$	$90.22_{\pm0.48}$
DMGI [90]	$88.83_{\pm0.48}$	$79.13_{\pm2.11}$	$96.07_{\pm1.89}$	$92.65_{\pm1.57}$	$93.83_{\pm1.67}$	$86.03_{\pm2.45}$	$93.11_{\pm0.98}$	$88.51_{\pm1.00}$	$93.87_{\pm0.84}$	$87.56_{\pm0.55}$
SSDCM [89]	$89.23_{\pm0.87}$	$80.84_{\pm2.32}$	$97.11_{\pm0.63}$	$94.82_{\pm0.96}$	$94.72_{\pm1.59}$	$86.69_{\pm0.76}$	$93.30_{\pm0.25}$	$89.22_{\pm1.94}$	$94.96_{\pm0.52}$	$88.61_{\pm0.38}$
MLP [86]	$80.74_{\pm0.43}$	$61.80_{\pm1.48}$	$93.01_{\pm0.60}$	$87.90_{\pm0.54}$	$88.78_{\pm0.44}$	$86.00_{\pm1.49}$	$93.12_{\pm0.64}$	$89.92_{\pm0.55}$	$94.59_{\pm0.94}$	$90.42_{\pm0.89}$
MStream [16]	$73.90_{\pm1.13}$	$70.22_{\pm1.61}$	$78.48_{\pm0.19}$	$74.04_{\pm1.66}$	$82.47_{\pm1.67}$	$77.89_{\pm0.58}$	$89.47_{\pm1.13}$	$84.38_{\pm1.01}$	$88.34_{\pm0.45}$	$83.66_{\pm1.79}$
LUCID [37]	$83.62_{\pm1.69}$	$72.31_{\pm1.14}$	$94.36_{\pm0.41}$	$89.46_{\pm0.72}$	$88.94_{\pm1.73}$	$85.23_{\pm0.94}$	$92.77_{\pm1.39}$	$88.32_{\pm0.91}$	$95.84_{\pm1.46}$	$90.75_{\pm0.79}$
Ours (DIDS)	$\mathbf{96.12_{\pm2.75}}$	$\mathbf{98.69_{\pm0.42}}$	$\mathbf{98.24_{\pm0.32}}$	$\mathbf{96.32_{\pm0.25}}$	$\mathbf{96.83_{\pm0.36}}$	$\mathbf{92.34_{\pm1.10}}$	$\mathbf{95.45_{\pm0.67}}$	$\mathbf{91.55_{\pm1.03}}$	$\mathbf{96.34_{\pm0.21}}$	$\mathbf{93.23_{\pm1.50}}$

(a) EdgeIIoT

(b) CIC-ToN-IoT

(c) CIC-BoT-IoT

(d) NF-UNSW-NB15-v2

(e) NF-CSE-CIC-IDS2018-v2

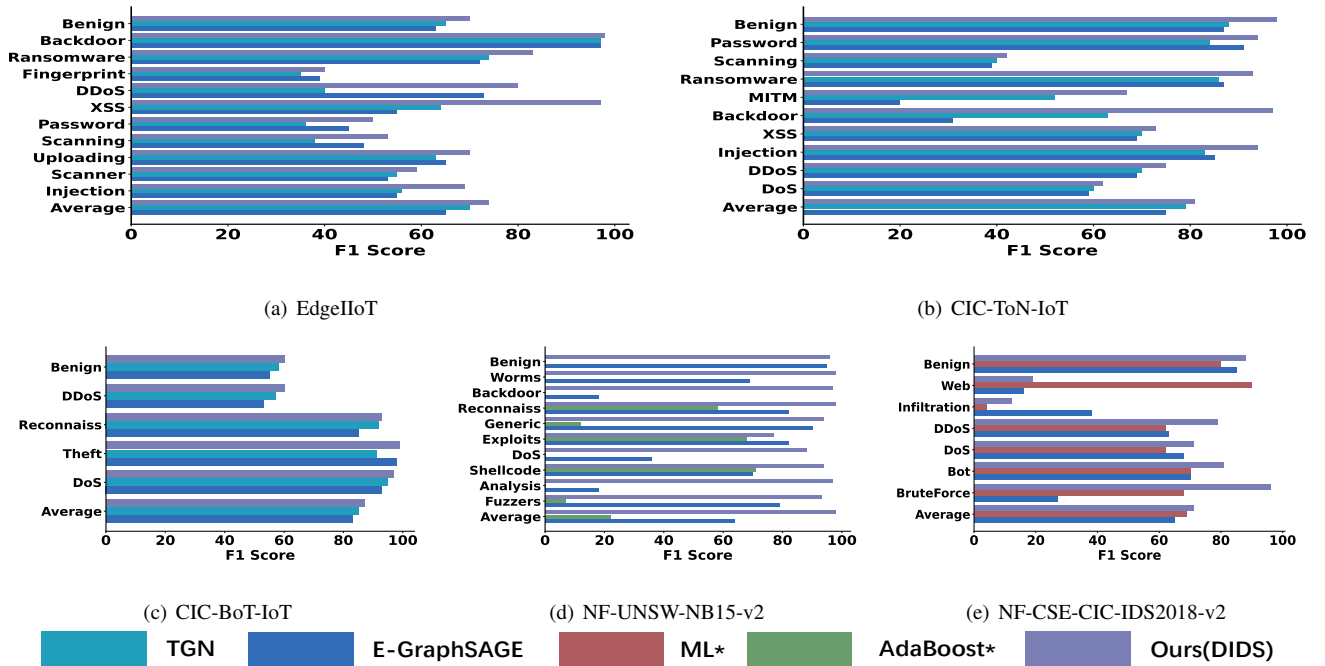TGN   E-GraphSAGE   ML⋆   AdaBoost⋆   Ours(DIDS)

Fig. 6. Comparisons of multi-classification. Here † indicates that the results are directly copied from the previous works.

disentanglements. More details about unknown detection are available in Appendix 17.1.

4) Comparisons under few-shot settings:

Task Setting: We first conduct the DIDS pretraining over the known benign and attack traffic. Then we conduct a $\mathcal{N}$-way-$\mathcal{K}$-shot learning for the few-shot attack traffic. Specifically, $\mathcal{N}$ is the category number of few-shot traffic, and $\mathcal{K}$ is the training number of each category, referred to support set. In our few-shot experiments, we set $\mathcal{K}$ as 5. We also construct a query set for our few-shot task, with 15 sample numbers for each category. The query set can be sampled from the category samples, or augmented from the support set. We split a 5-fold cross-validation set from the training set, with 20% of support and query set. Taking CIC-TON-IoT as an example, we conduct a five-way-five-shot learning and five-classification test with 100 training samples. We repeat the above training,

validate, test ten times, and report the average performance. Each time's training and validation samples are randomly selected from our existing traffic data, thus simulating the few-shot learning setting in real-world scenarios.

Baselines: We select 14 few-shot learning models to incorporate into DIDS as baselines, including 3 meta-learning based models (i.e., MBase [22], MTL [62], TEG [101]), where TEG is designed for graph-structure-based few-shot learning, 4 augmentation-based models (i.e., CLSA [71], ESPT [72], ICI [70], KSCL [73], where CLAS and ESPT are based on contrastive augmentation, ICI and KSCL are based on instance augmentation), 4 metric learning-based models (i.e., BSNet [27], CMFSL [31], TAD [32], PCWPK [102]), 3 selective and strong baselines in current few-shot intrusion detection area (i.e., FeCoGraph [103], FC-Net [104], and BSF-NID [105]).

Performance metrics: We follow the previous work [106]

TABLE II
COMPARISONS OF FEW-SHOT LEARNING CLASSIFICATION ON FIVE DATASETS. THE RESULTS ARE THE AVERAGE SCORES OF TEN-TIME REPETITIONS.

Methods	CIC-TON-IoT		CIC-BoT-IoT		EdgeIIoT		NF-UNSW-NB15-v2		NF-CSE-CIC-IDS2018-v2	
	F1	NMI	F1	NMI	F1	NMI	F1	NMI	F1	NMI
MBase [22]	$41.74\pm_{3.22}$	$35.81\pm_{2.84}$	$38.52\pm_{3.58}$	$27.89\pm_{3.84}$	$51.13\pm_{2.31}$	$55.61\pm_{2.05}$	$41.74\pm_{3.22}$	$35.81\pm_{2.84}$	$63.74\pm_{4.14}$	$68.76\pm_{3.16}$
MTL [62]	$30.79\pm_{3.13}$	$29.81\pm_{3.58}$	$37.80\pm_{5.66}$	$25.10\pm_{2.69}$	$52.64\pm_{5.69}$	$55.41\pm_{4.16}$	$44.18\pm_{4.22}$	$35.85\pm_{2.79}$	$45.38\pm_{5.88}$	$47.04\pm_{4.82}$
TEG [101]	$29.92\pm_{2.17}$	$22.64\pm_{2.22}$	$29.62\pm_{3.74}$	$25.71\pm_{2.82}$	$26.34\pm_{2.45}$	$27.58\pm_{2.64}$	$27.80\pm_{2.14}$	$21.44\pm_{1.74}$	$25.20\pm_{3.17}$	$23.48\pm_{3.97}$
CLSA [71]	$17.21\pm_{3.63}$	$45.01\pm_{2.83}$	$38.54\pm_{3.49}$	$51.70\pm_{4.90}$	$18.84\pm_{3.10}$	$44.25\pm_{3.27}$	$17.53\pm_{1.95}$	$45.67\pm_{4.13}$	$19.19\pm_{2.90}$	$49.76\pm_{3.46}$
ESPT [72]	$36.76\pm_{4.37}$	$38.51\pm_{3.29}$	$39.41\pm_{5.37}$	$34.66\pm_{3.49}$	$61.28\pm_{8.34}$	$53.77\pm_{6.91}$	$46.58\pm_{5.94}$	$49.37\pm_{6.58}$	$41.37\pm_{4.98}$	$46.52\pm_{6.71}$
ICI [70]	$55.67\pm_{5.14}$	$47.80\pm_{3.77}$	$71.32\pm_{3.35}$	$57.51\pm_{4.50}$	$49.74\pm_{4.65}$	$51.30\pm_{5.27}$	$39.17\pm_{3.66}$	$31.81\pm_{2.21}$	$76.53\pm_{9.43}$	$79.00\pm_{6.20}$
KSCL [73]	$36.19\pm_{2.66}$	$38.17\pm_{3.90}$	$33.50\pm_{4.23}$	$29.40\pm_{4.80}$	$34.87\pm_{2.14}$	$30.63\pm_{2.56}$	$17.03\pm_{1.97}$	$21.90\pm_{1.98}$	$25.74\pm_{3.01}$	$28.26\pm_{3.15}$
BSNet [27]	$58.41\pm_{2.36}$	$57.76\pm_{2.96}$	$42.00\pm_{5.19}$	$31.41\pm_{4.71}$	$43.74\pm_{4.20}$	$49.68\pm_{5.09}$	$65.73\pm_{2.65}$	$58.96\pm_{2.08}$	$56.39\pm_{5.75}$	$58.56\pm_{3.76}$
CMFSL [31]	$37.46\pm_{5.37}$	$47.51\pm_{4.29}$	$58.62\pm_{3.58}$	$51.47\pm_{6.28}$	$39.43\pm_{3.88}$	$43.67\pm_{5.75}$	$51.30\pm_{4.29}$	$47.62\pm_{6.47}$	$61.37\pm_{4.95}$	$58.42\pm_{6.17}$
TAD [32]	$49.80\pm_{2.13}$	$42.23\pm_{1.35}$	$62.75\pm_{6.40}$	$56.53\pm_{7.70}$	$50.82\pm_{3.93}$	$53.40\pm_{3.86}$	$58.97\pm_{3.76}$	$58.72\pm_{2.60}$	$54.88\pm_{3.34}$	$52.15\pm_{1.57}$
PCWPK [102]	$42.92\pm_{3.80}$	$44.17\pm_{3.12}$	$56.08\pm_{2.91}$	$53.01\pm_{5.17}$	$34.74\pm_{2.54}$	$34.29\pm_{3.36}$	$51.97\pm_{1.76}$	$43.62\pm_{2.27}$	$52.39\pm_{8.60}$	$64.56\pm_{5.57}$
FeCoGraph [103]	$45.78\pm_{1.96}$	$41.08\pm_{1.56}$	$43.24\pm_{0.88}$	$40.12\pm_{1.54}$	$28.66\pm_{6.66}$	$26.75\pm_{9.01}$	$34.51\pm_{1.02}$	$28.79\pm_{1.42}$	$64.85\pm_{16.04}$	$74.83\pm_{10.51}$
FC-Net [104]	$\underline{78.55}\pm_{2.30}$	$24.27\pm_{3.53}$	$\underline{76.60}\pm_{3.95}$	$27.04\pm_{5.29}$	$58.48\pm_{3.75}$	$1.07\pm_{0.67}$	$\underline{64.58}\pm_{3.31}$	$2.82\pm_{2.09}$	$\underline{77.48}\pm_{1.32}$	$22.32\pm_{1.82}$
BSF-NID [105]	$73.95\pm_{1.15}$	$\underline{66.97}\pm_{0.59}$	$73.21\pm_{0.05}$	$\underline{69.44}\pm_{0.32}$	$51.21\pm_{0.16}$	$54.68\pm_{0.64}$	$35.73\pm_{1.50}$	$45.61\pm_{0.76}$	$68.56\pm_{1.02}$	$67.05\pm_{1.53}$
Ours(DIDS-MFL)	$\mathbf{97.47}\pm_{1.17}$	$\mathbf{94.27}\pm_{2.62}$	$\mathbf{86.11}\pm_{0.84}$	$\mathbf{85.36}\pm_{4.25}$	$\mathbf{92.73}\pm_{2.57}$	$\mathbf{88.21}\pm_{3.25}$	$\mathbf{96.00}\pm_{1.12}$	$\mathbf{92.04}\pm_{1.36}$	$\mathbf{93.93}\pm_{3.27}$	$\mathbf{90.98}\pm_{3.69}$

TABLE III
UNKNOWN CLASSIFICATION ON THE CIC-TON-IoT DATASET WITH
METRIC F1-SCORE (%)

Method	Logistic Regression	MSteam	E-GraphSAGE	TGN	Ours (DIDS)
DDoS	$1.68\pm_{0.67}$	$26.73\pm_{1.21}$	$10.20\pm_{1.46}$	$32.84\pm_{1.03}$	$\mathbf{41.78}\pm_{0.26}$
MITM	$2.17\pm_{0.84}$	$12.82\pm_{1.90}$	$6.05\pm_{0.35}$	$15.32\pm_{0.54}$	$\mathbf{34.91}\pm_{0.91}$
Injection	$0.00\pm_{0.34}$	$15.73\pm_{1.73}$	$12.37\pm_{0.88}$	$22.83\pm_{0.35}$	$\mathbf{25.63}\pm_{0.93}$
Backdoor	$3.13\pm_{1.33}$	$20.85\pm_{0.63}$	$9.51\pm_{0.46}$	$23.10\pm_{1.03}$	$\mathbf{32.29}\pm_{0.81}$

using F1 scores and NMI [107] metrics for multi-classification comparison.

As shown in Table II, our MFL consistently achieves the best performances among 14 baselines under the five benchmarks. Specifically, our MFL achieves reproducible average results of 93.25% and 90.17% in F1 score and NMI value over five public datasets. The results gain 71.91% - 125.19% and 71.95% - 144.07% improvements in average F1-score and NMI value of 14 baselines over five public datasets, respectively. Furthermore, compared to the previous state-of-the-art approaches, the proposed DIDS-MFL still achieves a 12.42%-51.32% and 15.16%-58.62% improvement in F1 score and NMI, respectively. Specifically, the selective baselines in few-shot intrusion detection achieve better performance compared to the previous approaches, such as FC-Net, achieving the second-best F1 score of 78.55% in CIC-ToN-IoT, and BSF-NID, achieving the second-best NMI value of 69.44% in CIC-BOT-IoT. However, these strong baselines are still significantly lower than the proposed DIDS-MFL, lying in two aspects: 1) The benchmarks cover more comprehensive real-world traffic data, including malicious traffic data with complex patterns. It poses a critical challenge to the existing baselines in detecting some complex attacks, especially under few-shot intrusion detection settings. 2) The existing few-shot intrusion detection approaches can hardly separate the few-shot representations in the latent space and highlight the few-shot attack-specific information, thus misleading the few-shot threat detection. It manifested that the proposed MFL performs well in both F1 score and NMI value, while existing approaches, e.g., FC-Net, perform worse in NMI value, a stricter metric to evaluate multi-classification performance. We attribute the superior

and impressive reproducible results of MFL to our multi-scale latent information learning and disentanglement designs among traffic representation dimensions. The $RQ5$ and $RQ6$ in the discussion section further verified our attributions.

C. Ablation Study

1) DIDS: In this section, we conduct an ablation study on the CIC-ToN-IoT dataset to evaluate the effectiveness of each component. We remove our statistical disentanglement and denote it as "w/o SD". We use "w/o RD" and "w/o ML-GRAND" to refer to the model that removes representational disentanglement and the multi-layer graph diffusion module, respectively. Table IV reports the comparison results in binary classification. It shows that removing the multi-layer graph diffusion module leads to the most significant performance degradation, e.g., an 18.33 points decrease in AUC, indicating that it is the key component for the accuracy of the proposed DIDS. Our second disentangled memory is also non-trivial to the overall detection accuracy, as removing this component can decrease the performance by 12.47 points in AUC. We observe that the SD module also benefits the model performance. The above ablation study further confirms the effectiveness of the three key components.

TABLE IV
ABLATION STUDY OF DIDS. F1-SCORE (%), PRECISION (%), ROC-AUC (%), RECALL (%), ALL METRICS ABOVE ARE THE AVERAGE OF FIVE REPEATED EXPERIMENTS ON THE CIC-TON-IoT DATASET.

Variants	P	R	F1	AUC
w/o SD	$92.70\pm_{0.46}$	$90.71\pm_{0.89}$	$91.69\pm_{0.33}$	$86.87\pm_{0.54}$
w/o RD	$91.06\pm_{0.57}$	$87.32\pm_{0.67}$	$89.15\pm_{0.42}$	$83.57\pm_{1.33}$
w/o MLGRAND	$88.76\pm_{0.54}$	$84.43\pm_{0.26}$	$86.54\pm_{0.71}$	$79.32\pm_{0.30}$
DIDS(ours)	$\mathbf{97.78}\pm_{0.32}$	$\mathbf{98.06}\pm_{0.43}$	$\mathbf{97.92}\pm_{0.26}$	$\mathbf{96.04}\pm_{0.25}$

2) MFL: Furthermore, we conduct an ablation study on the CIC-TON-IOT dataset to evaluate the effectiveness of our designed multi-scale few-shot learning (MFL) module. We denote our model without multi-scale latent optimization space as "w/o LOS" and "w/o DR" as our model without the

disentanglement regularization term. For the model without both of the above components, we denote it as "SE", which degenerated into a naive self-expressiveness model. Table V shows that removing both LOS and DR leads to dramatic performance degradation, i.e., 9.92% and 9.75% drop in F1-score and NMI, respectively, indicating the effectiveness of our designed MFL. Furthermore, we observed that each component of MFL, e,g. LOS or DR is necessary for performance improvement, cause removing one of them will lead to performance degradation, i.e., 22.21%-23.16% and 8.63%-9.24% drop in F1-score and NMI, respectively. Without any one of these two components, the performance will be poorer than the base model SE. Results showcase the complementary relationships of the proposed two modules. The LOS provides a latent optimization space across multi-scale representations, which is the foundation of the DR term. Then the DR term generates effective disentangled representations across multiple scales based on LOS.

TABLE V
ABLATION STUDY OF MFL. F1-SCORE (%), PRECISION (%), ROC-AUC (%), RECALL (%), ALL METRICS ABOVE ARE THE AVERAGE OF TEN REPEATED EXPERIMENTS ON THE CIC-TON-IOT DATASET.

Variants	P	R	F1	NMI
w/o LOS	$77.80\pm_{3.83}$	$78.80\pm_{1.99}$	$75.82\pm_{2.89}$	$86.13\pm_{2.15}$
w/o DR	$75.51\pm_{3.01}$	$77.30\pm_{1.77}$	$74.90\pm_{1.66}$	$85.56\pm_{1.64}$
SE	$88.62\pm_{3.56}$	$88.80\pm_{3.27}$	$87.80\pm_{3.17}$	$85.08\pm_{2.31}$
MFL(ours)	**$97.68\pm_{1.09}$**	**$97.50\pm_{1.18}$**	**$97.47\pm_{1.17}$**	**$94.27\pm_{2.62}$**

D. Discussion

1) DIDS

RQ1: How does the statistical disentanglement help the detection of various attacks? To answer this question, we visualize the distributions of features before and after the statistical disentanglement. Figure 7 shows the visualizations of the two distributions respectively. We can observe that there is less overlap between distributions of features after the disentanglement compared with the original data, which demonstrates this module could decrease the mutual reference between features and enable them to be distinguishable. We also observe that the distributions gradually shift to the right side, representing the order-preserved constraints within our disentangling method.
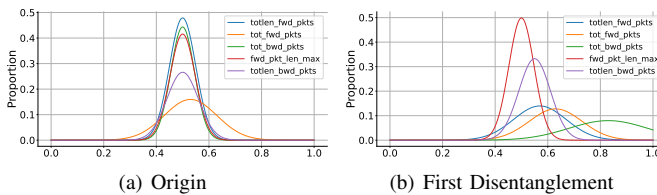

Fig. 7. Statistical disentanglement of traffic features.

RQ2: How does the representational disentanglement benefit "highlighting the attack-specific features"? To answer this question, we track several Injection attack data in the CIC-ToN-IoT dataset and obtain the representation of these data in

DIDS and E-GraphSAGE. Meanwhile, we calculate the above two methods' average values of embeddings for the benign traffic. As shown in Figure 8, the representation values of E-GraphSAGE are much closer to the normal. It illustrates that as nodes aggregate, the discrepancies in features become blurred, leading to inaccurate classification. While benefiting from the representational disentanglement, each dimension of features in DIDS can effectively preserve its own properties, deviating from the averages. Especially for the attack-specific features Fwd Pkt Len Max and Fwd Pkt Std, etc., these attack-specific features are significantly highlighted in Fig. 1(a), thus improving the accuracy of detection. The result proves the effectiveness of the proposed DIDS in maintaining a disentangled representation during the aggregation process, ensuring the presence of discrepancies, thus highlighting the attack-specific features and leading to more accurate classification for attacks.
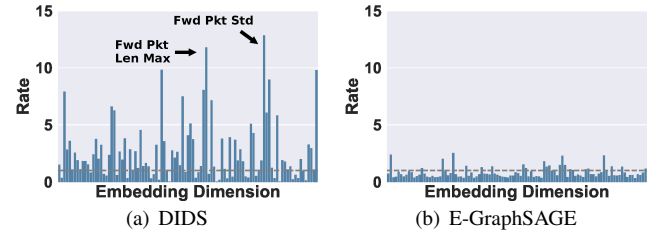

Fig. 8. The comparison of node representation of the Injection attack after graph aggregation of our DIDS and E-GraphSAGE. The grey line presents benign data.

RQ3: How does the multi-layer diffusion module perform effectively for intrusion detection? We have illustrated the principle of multi-layer diffusion in Section IV-C. In this part, we take the MITM attack as an example to illustrate the effectiveness of spatial-temporal in intrusion detection.
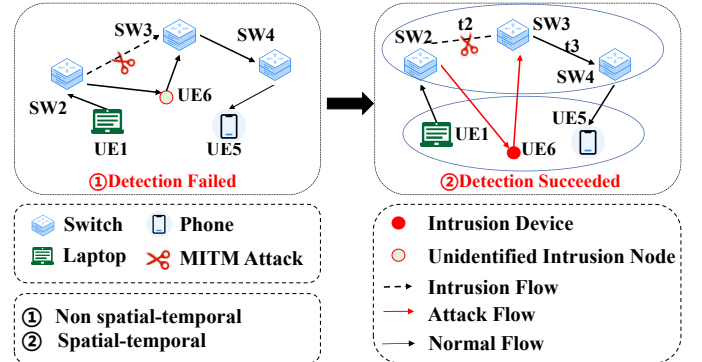

Fig. 9. Spatial-temporal coupling in intrusion detection.

Figure 9 (a) shows a deep learning-based NIDS. When a MITM attack occurs, it is difficult to detect the intrusion since the spatial and temporal information of those packets is not considered. There are also some methods that only consider a single aspect of spatial and temporal information, such as E-GraphSAGE and MStream. In this case, for example, E-GraphSAGE mainly focuses on the spatial relationship of the set nodes and extracts features from them. However, we observe that different streams have their own timestamps

from Table VI, so the lack of temporal information makes it impossible to analyze the dynamic structural changes of the edge. Similarly, taking the temporal information as the only effect factor will also get incomplete characteristics that do not contain spatial information (IP address). Moreover, some methods that take both the spatial and temporal information into account, such as Euler, take the snapshot method to capture the feature of the flow which does not achieve the synchronous update for spatial and temporal information. As shown in Figure 9 (b), intuitively, we can quickly detect that UE6 is an intrusion device of layer 1 when the flow changes from $SW2 - SW3$ to $SW2 - UE6$ and $UE6 - SW3$ considering $SW2$, $SW3$ are layer 2 devices. Also, we have noticed the changes in dynamic graph structure with a multi-Layer graph diffusion module to realize spatio-temporal coupling and synchronous updating. Overall, DIDS performs best among these baselines in detecting various attacks.

TABLE VI
TIMESTAMP AND IP

Time	TimeStamp	Src IP	Dst IP
$t1$	25/04/2019 05 : 18 : 37 pm	183.68.192.168	1.169.216.58
$t2$	25/04/2019 05 : 18 : 42 pm	1.169.216.58	25.162.192.168
$t3$	25/04/2019 05 : 18 : 42 pm	1.169.216.58	230.158.52.59
$t4$	25/04/2019 05 : 18 : 49 pm	230.158.52.59	25.162.192.168
$t5$	25/04/2019 05 : 18 : 52 pm	25.162.192.168	69.151.192.168
$t6$	25/04/2019 05 : 19 : 00 pm	177.21.192.168	230.158.52.59

RQ4**: How does the disentanglement facilitate the explainability of DIDS?** For this question, we rely on Figure 8 (a) as an example to recover the possible traffic features of a password attack. Since the original features are retained after disentanglement, we can find some feature values that deviate significantly from the normal values in node embeddings. In the password attack, we observe that the deviated features after disentanglement are "Fwd Pkt Std" and "Fwd Pkt Len Max". It aligns with our common sense for the main causes of password attacks and further benefits the explainability of DIDS.

2) MFL
RQ5**: How does multi-scale transform-based MFL benefit the distinction of few-shot traffic threats?** To answer this question, we visualize the learned representations of DIDS and the few-shot learning module MFL on the CIC-TON-IoT dataset via t-SNE technology. As shown in Fig. 10(a) (b), the MFL's representations are highly separated and distinguishable for different few-shot attacks, i.e., the red square box, compared to the representation generated by DIDS, i.e., the red round box. It verifies the effectiveness of our proposed multi-scale few-shot learning framework, i.e., multiple coefficient matrices fusion and multi-scale transformation. They discover the attack-specific invariant features among few-shot traffic in latent space, thus improving the distinction of attack representations.
RQ6**: Can MFL disentangle the representations of few-shot samples?** To answer this question, we visualize the learned representations of DIDS and MFL on the CIC-TON-IoT dataset via correlation heatmaps. As shown in Fig. 11(a) (b), the visualization results significantly reveal that MFL can generate highly disentangled representations via our designed

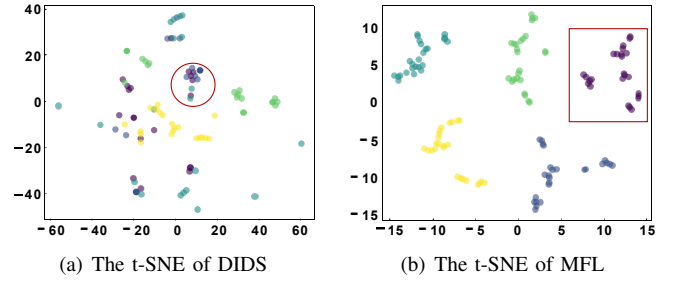(a) The t-SNE of DIDS   (b) The t-SNE of MFL

Fig. 10. The t-SNE visualization of representations on the CIC-TON-IoT dataset generated by DIDS and MFL.

regularization term. Specifically, MFL generates a block diagonal heatmap with high correlations and non-diagonal areas with very low correlations. It verifies that MFL disentangles the few-shot traffic representations and highlights the attack-specific ones, making them more distinguishable.
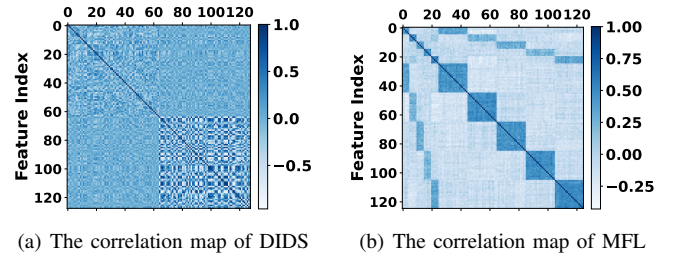

(a) The correlation map of DIDS   (b) The correlation map of MFL

Fig. 11. The correlation map and the t-SNE visualization of representations generated by DIDS and DIDS-MFL with MFL.

RQ8**: Can MFL improve the performance of DIDS?** So far, we have verified the effectiveness of MFL in few-shot traffic intrusion detection, including serving as a plug-and-play module for other methods. To further study the applicability of MFL in supervised task, we conduct DIDS training with MFL and report the multi-classification results, as shown in Table VII. DIDS with MFL achieves a 2.31% to 15.82% F1-score improvements among three datasets. The results reveal the effectiveness of MFL when serving as a multi-classification module for normal-size traffic training. Our designed multi-scale transform-based framework also sheds light on the future intrusion detection model design. We also demonstrate the effectiveness of the proposed MFL module under multiple baselines in few-shot intrusion detection tasks and their visualization analyses in Appendix 20 and 21, respectively.

TABLE VII
THE MULTI-CLASSIFICATION F1-SCORE (%) COMPARISON BETWEEN DIFFERENT DATASETS ON DIDS WITH MFL MODULE.

Datasets	CIC-TON-IoT	CIC-BoT-IoT	NF-UNSW-NB15-v2
DIDS	83.56 ± 1.18	82.98 ± 1.45	93.55 ± 0.97
DIDS-MFL	$\mathbf{96.78 \pm 1.72}$	$\mathbf{85.72 \pm 0.56}$	$\mathbf{95.71 \pm 5.89}$

RQ9**: Network intrusion detection and Large language model (LLM).** Recently, there has been a significant surge in the development and application of Large Language Models (LLMs) [108] cross various domains, including Natural

Language Processing (NLP) [109], Computer Vision (CV) [110], and multimodal tasks. However, the existing LLMs, e.g. GPT-3.5, GPT-4 struggle to detect traffic threats accurately and efficiently. As shown in Table VIII, the F1-score of the existing LLM is significantly lower than our SOTA method DIDS. It may lie in the hallucinations [111] of existing neural language-based LLM, which struggle to comprehend the intrinsic attack features in traffic data. Second, the LLM necessitates more time to process the input traffic data and may fail as the size of the input traffic increases. It may be attributed to the numerical values of the traffic data involved. To sum up, the future directions to empower NIDS via LLM can be: 1. Aligning encrypted traffic data with the input requirements of LLMs, thus speeding the intrusion detection via LLM; 2. Determining the necessary volume of traffic data and corresponding traffic mining technology to unlock the powerful inference capabilities of LLM.

TABLE VIII
THE F1-SCORE (%) AND TIME COST (S) COMPARISONS BETWEEN THE LLM AND DIDS-MFL ON 100 TRAFFIC SAMPLES FROM THE CIC-TON-IoT DATASET.

	F1-Score	Time Cost
LLM	66.67%	13.88s
DIDS-MFL	**99.52%**	**6.41**s

VI. CONCLUSION

This paper quantitatively discovers the inconsistent performances of existing NIDS and reveals that the underlying cause is entangled feature distributions. Furthermore, we delve into the deeper reasons for the poor few-shot intrusion detection performance of existing NIDS. These interesting observations motivate us to propose DIDS-MFL. The former is a novel method that aims to benefit known and unknown attacks with a double disentanglement scheme and graph diffusion mechanism, and the latter is a transform-based multi-scale few-shot learner to highlight few-shot traffic threats. The proposed DIDS first employs statistical disentanglement on the traffic features to automatically differentiate tens and hundreds of complex features and then employs representational disentanglement on the embeddings to highlight attack-specific features. DIDS also fuses the network topology via multi-layer graph diffusion methods for dynamic intrusion detection. Finally, the proposed MFL uses an alternating optimization framework to separate and disentangle the few-shot traffic representations. Extensive experiments on five benchmarks show the effectiveness and the practical employment potential of our DIDS-MFL, including binary classifications, multi-classifications, unknown attack detection, and few-shot intrusion detection. Future work could focus on the deployment of DIDS-MFL on future wireless networks.

REFERENCES

[1] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *2012 IEEE symposium on security and privacy*. IEEE, 2012, pp. 523–537.

[2] Z. Cekerevac, Z. Dvorak, L. Prigoda, and P. Cekerevac, "Internet of things and the man-in-the-middle attacks–security and economic risks," *MEST Journal*, vol. 5, no. 2, pp. 15–25, 2017.

[3] K. M. Prasad, "Dos and ddos attacks: defense, detection and trace-back mechanisms-a survey," *Global Journal of Computer Science and Technology*, vol. 14, no. E7, pp. 15–32, 2014.

[4] B. Bhushan and G. Sahoo, "Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks," *Wireless Personal Communications*, vol. 98, pp. 2037–2077, 2018.

[5] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 147–152.

[6] C. Wueest, "Targeted attacks against the energy sector," *Symantec Security Response, Mountain View, CA*, 2014.

[7] S. Kumar, A. Viinikainen, and T. Hamalainen, "Machine learning classification model for network based intrusion detection system," in *2016 11th international conference for internet technology and secured transactions (ICITST)*. IEEE, 2016, pp. 242–249.

[8] H. Zhao, N. Zheng, J. Li, J. Yao, and Q. Hou, "Unknown malware detection based on the full virtualization and svm," in *2009 International Conference on Management of e-Commerce and e-Government*. IEEE, 2009, pp. 473–476.

[9] P. Ioulianou, V. Vasilakis, I. Moscholios, and M. Logothetis, "A signature-based intrusion detection system for the internet of things," *Information and Communication Technology Form*, 2018.

[10] F. Erlacher and F. Dressler, "Fixids: A high-speed signature-based flow intrusion detection system," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–8.

[11] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based intrusion detection systems," *Applied Soft Computing*, vol. 92, p. 106301, 2020.

[12] N. T. Van, T. N. Thinh *et al.*, "An anomaly-based network intrusion detection system using deep learning," in *2017 international conference on system science and engineering (ICSSE)*. IEEE, 2017, pp. 210–214.

[13] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[14] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[15] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.

[16] S. Bhatia, A. Jain, P. Li, R. Kumar, and B. Hooi, "MStream: Fast anomaly detection in multi-aspect streams," in *Proceedings of the Web Conference 2021*. ACM, apr 2021. [Online]. Available: https://doi.org/10.1145%2F3442381.3450023

[17] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-graphsage: A graph neural network based intrusion detection system for iot," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[18] C. Ioannou and V. Vassiliou, "Network attack classification in iot using support vector machines," *Journal of Sensor and Actuator Networks*, vol. 10, no. 3, p. 58, 2021.

[19] N. Moustafa, "A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets," *Sustainable Cities and Society*, vol. 72, p. 102994, 2021.

[20] C. Qiu, Y. Geng, J. Lu, K. Chen, S. Zhu, Y. Su, G. Nan, C. Zhang, J. Fu, Q. Cui *et al.*, "3d-ids: Doubly disentangled dynamic intrusion detection," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1965–1977.

[21] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.

[22] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," 2021.

[23] X. Li, Z. Zhang, X. Tan, C. Chen, Y. Qu, Y. Xie, and L. Ma, "Promptad: Learning prompts with only normal samples for few-shot anomaly detection," 2024.

[24] N. Belton, M. T. Hagos, A. Lawlor, and K. M. Curran, "Fewsome: One-class few shot anomaly detection with siamese networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 2978–2987.

[25] J. Liao, X. Xu, M. C. Nguyen, A. Goodge, and C. S. Foo, "Coft-ad: Contrastive fine-tuning for few-shot anomaly detection," 2024.

[26] Y. Ma, Z. Wang, Y. Cao, and A. Sun, "Few-shot event detection: An empirical study and a unified view," 2023.

[27] X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, and J.-H. Xue, "Bsnet: Bi-similarity network for few-shot fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 30, p. 1318–1331, 2021. [Online]. Available: http://dx.doi.org/10.1109/TIP.2020.3043128

[28] G. Liu and J. Guo, "Bidirectional lstm with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[29] C. Brodbeck and J. Z. Simon, "Continuous speech processing," *Current Opinion in Physiology*, vol. 18, pp. 25–31, 2020.

[30] V. Sharma and R. N. Mir, "A comprehensive and systematic look up into deep learning based object detection techniques: A review," *Computer Science Review*, vol. 38, p. 100301, 2020.

[31] B. Xi, J. Li, Y. Li, R. Song, D. Hong, and J. Chanussot, "Few-shot learning with class-covariance metric for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 5079–5092, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251068999

[32] M. Hu, H. Chang, Z. Guo, B. Ma, S. Shan, and X. CHEN, "Understanding few-shot learning: Measuring task relatedness and adaptation difficulty via attributes," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=Pvgxecj5aS

[33] R. Samrin and D. Vasumathi, "Review on anomaly based network intrusion detection system," in *2017 international conference on electrical, electronics, communication, computer, and optimization techniques (ICEECCOT)*. IEEE, 2017, pp. 141–147.

[34] L. Ma, Y. Chai, L. Cui, D. Ma, Y. Fu, and A. Xiao, "A deep learning-based ddos detection framework for internet of things," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[35] N. Hu, Z. Tian, H. Lu, X. Du, and M. Guizani, "A multiple-kernel clustering based intrusion detection scheme for 5g and iot networks," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3129–3144, 2021.

[36] Y. Li, R. Li, Z. Zhou, J. Guo, W. Yang, M. Du, and Q. Liu, "Graphddos: Effective ddos attack detection using graph neural networks," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2022, pp. 1275–1280.

[37] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for ddos attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, 2020.

[38] M. Kalander, M. Zhou, C. Zhang, H. Yi, and L. Pan, "Spatio-temporal hybrid graph convolutional network for traffic forecasting in telecommunication networks," *arXiv preprint arXiv:2009.09849*, 2020.

[39] W. Cong, Y. Wu, Y. Tian, M. Gu, Y. Xia, M. Mahdavi, and C.-c. J. Chen, "Dynamic graph representation learning via graph transformer networks," *arXiv preprint arXiv:2111.10447*, 2021.

[40] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 914–921.

[41] I. J. King and H. H. Huang, "Euler: Detecting network lateral movement via scalable temporal link prediction," *Network and Distributed Systems Security (NDSS) Symposium*, 2022.

[42] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An approach to evaluating interpretability of machine learning," *arXiv preprint arXiv:1806.00069*, p. 118, 2018.

[43] M. W. Gondal, M. Wüthrich, Đ. Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer, "On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset," *Advances in Neural Information Processing Systems 32*, vol. 20, pp. 15 661–15 672, 2020.

[44] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen, "Weakly-supervised disentanglement without compromises," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6348–6359.

[45] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[46] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International conference on learning representations*, 2017.

[47] H. Kim and A. Mnih, "Disentangling by factorising," 2019.

[48] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang, "Causalvae: Structured causal disentanglement in variational autoencoder," 2023.

[49] R. Suter, D. Miladinovic, B. Schölkopf, and S. Bauer, "Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6056–6065.

[50] K. Ridgeway and M. C. Mozer, "Learning deep disentangled embeddings with the f-statistic loss," *Advances in neural information processing systems*, vol. 31, 2018.

[51] F. Träuble, E. Creager, N. Kilbertus, F. Locatello, A. Dittadi, A. Goyal, B. Schölkopf, and S. Bauer, "On disentangled representations learned from correlated data," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 401–10 412.

[52] S. Zhou, Z. Guo, C. Aggarwal, X. Zhang, and S. Wang, "Link prediction on heterophilic graphs via disentangled representation learning," *arXiv preprint arXiv:2208.01820*, 2022.

[53] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 5363–5370.

[54] M. Kalander, M. Zhou, C. Zhang, H. Yi, and L. Pan, "Spatio-temporal hybrid graph convolutional network for traffic forecasting in telecommunication networks," *arXiv preprint arXiv:2009.09849*, 2020.

[55] F. Gu, H. Chang, W. Zhu, S. Sojoudi, and L. El Ghaoui, "Implicit graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 984–11 995, 2020.

[56] Q. Chen, Y. Wang, Y. Wang, J. Yang, and Z. Lin, "Optimization-induced graph implicit nonlinear diffusion," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3648–3661.

[57] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, "Grand: Graph neural diffusion," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1407–1418.

[58] J. Li, Y. Liu, and L. Zou, "Dyngcn: A dynamic graph convolutional network based on spatial-temporal modeling," in *Web Information Systems Engineering – WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 83–95. [Online]. Available: https://doi.org/10.1007/978-3-030-62005-9_7

[59] Z. Yue, H. Zhang, Q. Sun, and X.-S. Hua, "Interventional few-shot learning," 2020.

[60] D. Wang, Y. Deng, Z. Yin, H.-Y. Shum, and B. Wang, "Progressive disentangled representation learning for fine-grained controllable talking head synthesis," 2022.

[61] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[62] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," 2019.

[63] K. Singh and D. Malhotra, "Meta-Learning based efficient framework for diagnosing rare disorders: A comprehensive survey," *AIP Conference Proceedings*, vol. 3072, no. 1, p. 040003, 03 2024. [Online]. Available: https://doi.org/10.1063/5.0199881

[64] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9714–9723.

[65] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3540–3552, 2020.

[66] Y. Ouyang, B. Li, Q. Kong, H. Song, and T. Li, "Fs-ids: A novel few-shot learning based intrusion detection system for scada networks," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[67] Z. Shi, M. Xing, J. Zhang, and B. Hao Wu, "Few-shot network intrusion detection based on model-agnostic meta-learning with l2f method," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.

[68] Y. Yan, Y. Yang, F. Shen, M. Gao, and Y. Gu, "Meta learning-based few-shot intrusion detection for 5g-enabled industrial internet," *Complex & Intelligent Systems*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268699862

[69] W. Cho and E. Kim, "Improving augmentation efficiency for few-shot learning," *IEEE Access*, vol. 10, pp. 17 697–17 706, 2022.

[70] Y. Wang, L. Zhang, Y. Yao, and Y. Fu, "How to trust unlabeled data? instance credibility inference for few-shot learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6240–6253, 2022.

[71] X. Wang and G.-J. Qi, "Contrastive learning with stronger augmentations," 2022.

[72] Y. Rong, X. Lu, Z. Sun, Y. Chen, and S. Xiong, "Espt: A self-supervised episodic spatial pretext task for improving few-shot learning," 2023.

[73] H. Xu, H. Xiong, and G.-J. Qi, "K-shot contrastive learning of visual features with multiple instance augmentations," 2021.

[74] M. N. Rizve, S. Khan, F. S. Khan, and M. Shah, "Exploring complementary strengths of invariant and equivariant representations for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 836–10 846.

[75] B. Xi, J. Li, Y. Li, R. Song, D. Hong, and J. Chanussot, "Few-shot learning with class-covariance metric for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 5079–5092, 2022.

[76] Y. Zhou, J. Hao, S. Huo, B. Wang, L. Ge, and S.-Y. Kung, "Automatic metric search for few-shot learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[77] X. Chen, G. Zhu, and J. Wei, "Mmml: Multimanifold metric learning for few-shot remote-sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023.

[78] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4080–4090.

[79] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:4412459

[80] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 3637–3645.

[81] J. Oh, S. Kim, N. Ho, J.-H. Kim, H. Song, and S.-Y. Yun, "Understanding cross-domain few-shot learning based on domain similarity and few-shot difficulty," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=rH-X09cB50f

[82] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems: 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings 14*. Springer, 2008, pp. 337–340.

[83] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[84] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," *arXiv preprint arXiv:2002.07962*, 2020.

[85] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[86] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in iot networks," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*. IEEE, 2019, pp. 0452–0457.

[87] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[88] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[89] A. Mitra, P. Vijayan, R. Sanasam, D. Goswami, S. Parthasarathy, and B. Ravindran, "Semi-supervised deep learning for multiplex networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1234–1244.

[90] C. Park, D. Kim, J. Han, and H. Yu, "Unsupervised attributed multiplex network embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 5371–5378.

[91] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.

[92] M. Yoon, B. Hooi, K. Shin, and C. Faloutsos, "Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach," *ACM*, 2019.

[93] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems," in *Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10*. Springer, 2021, pp. 117–135.

[94] W. Lalouani and M. Younis, "Robust distributed intrusion detection system for edge of things," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 01–06.

[95] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ml-based network intrusion detection," *Big Data Research*, vol. 30, p. 100359, 2022.

[96] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-e: A self-supervised network intrusion detection system based on graph neural networks," *Knowledge-Based Systems*, vol. 258, p. 110030, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122011236

[97] X. Guo, B. Zhou, and S. Skiena, "Subset node anomaly tracking over large dynamic graphs," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD '22. Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3534678.3539389

[98] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "Machine learning basics," *Deep learning*, pp. 98–164, 2016.

[99] R. E. Schapire, "Explaining adaboost," *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52, 2013.

[100] R. S. Fisher, J. H. Cross, J. A. French, N. Higurashi, E. Hirsch, F. E. Jansen, L. Lagae, S. L. Moshé, J. Peltola, E. Roulet Perez *et al.*, "Operational classification of seizure types by the international league against epilepsy: Position paper of the ilae commission for classification and terminology," *Epilepsia*, vol. 58, no. 4, pp. 522–530, 2017.

[101] S. Kim, J. Lee, N. Lee, W. Kim, S. Choi, and C. Park, "Task-equivariant graph few-shot learning," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1120–1131.

[102] B. Zhang, X. Li, Y. Ye, and S. Feng, "Prototype completion for few-shot learning," 2021.

[103] Q. Mao, X. Lin, W. Xu, Y. Qi, X. Su, G. Li, and J. Li, "Fecograph: Label-aware federated graph contrastive learning for few-shot network intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2266–2280, 2025.

[104] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3540–3552, 2020.

[105] X. Dong, Y. Lai, X. Zhang, and X. Xu, "Counteracting new attacks in cps: A few-shot class-incremental adaptation strategy for intrusion detection system," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2025.

[106] S. Bandyopadhyay and V. Peter, "Unsupervised constrained community detection via self-expressive graph neural network," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 1078–1088.

[107] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020.

[108] A. Bahrini, M. Khamoshifar, H. Abbasimehr, R. J. Riggs, M. Esmaeili, R. M. Majdabadkohne, and M. Pasehvar, "Chatgpt: Applications, opportunities, and threats," 2023.

[109] I. Lauriola, A. Lavelli, and F. Aiolli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, 2022.

[110] F. Liu, D. Chen, F. Wang, Z. Li, and F. Xu, "Deep learning based single sample face recognition: a survey," *Artificial Intelligence*

Review, vol. 56, no. 3, p. 2723–2748, Aug. 2022. [Online]. Available: http://dx.doi.org/10.1007/s10462-022-10240-2

[111] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," 2023.

Chenyang Qiu received his B.Sc. degree in science from Beijing Jiaotong University, Beijing, China, in 2020. He is currently pursuing his Ph.D degree at the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications, Beijing. His research interests include Wireless Network Intrusion Detection, Semantic Communication, Graph Neural Networks, and AI Robustness. He has published papers in top-tier conferences and journals, including KDD, AAAI, IEEE Transactions on Mobile Computing, and IEEE Transactions on Cybernetics.

Guoshun Nan (Member, IEEE) is a professor at Beijing University of Posts and Telecommunications. He is a member of the National Engineering Research Center for Mobile Network Technologies. He has broad interest in natural language processing, computer vision, machine learning, and wireless communications, such as information extraction, model robustness, multimodal retrieval, and next-generation wireless networks. He has published papers in top-tier conferences and journals including ACL, CVPR, EMNLP, SIGIR, SIGCOMM, IEEE JSAC, IEEE Network, Computer Networks, Journal of Network, and Computer Applications. He served as a Reviewer for ACL, EMNLP, AAAI, Neurocomputing, and IEEE Transactions on Image Processing.

Hongrui Xia is an undergraduate student at the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications. He specializes in network security and intrusion detection, aiming to advance his expertise in safeguarding digital systems and detecting cyber threats. Dedicated to contributing to the field through research and innovation.

Zheng Weng is an undergraduate student at the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications. He focuses on Graph Neural Networks (GNNs) and machine learning, exploring innovative methods to enhance data security and network integrity through advanced computational models.

Xueting Wang is an undergraduate student at the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications. Her current research interests include artificial intelligence security, the analysis and detection of network traffic attacks, and the use of AI automation to detect and analyze viruses such as Trojans.

Meng Shen (Member, IEEE) received the B.Eng degree from Shandong University, Jinan, China in 2009, and the Ph.D degree from Tsinghua University, Beijing, China in 2014, both in computer science. Currently he serves in Beijing Institute of Technology, Beijing, China, as a professor. His research interests include privacy protection for cloud and IoT, blockchain applications, and encrypted traffic classification. He received the Best Paper Runner-Up Award at IEEE IPCCC 2014. He is a member of the IEEE.

Xiaofeng Tao (Senior Member, IEEE) received the bachelor's degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1993, and the master's and Ph.D. degrees in telecommunication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999 and 2002, respectively. He is currently a Professor at the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications, a fellow of the Institution of Engineering and Technology, and the Chair of the IEEE ComSoc Beijing Chapter. He has authored or coauthored over 200 articles and three books in wireless communication areas. He focuses on B5G/6G research.

Jun Liu (Member, IEEE) received the B.Eng. degree from Central South University in 2011, the M.Sc. degree from Fudan University in 2014, and the Ph.D. degree from Nanyang Technological University in 2019. He was with the Singapore University of Technology and Design from 2019 to 2024 and Tencent from 2014 to 2015. He is a Professor and the Chair of Digital Health at the School of Computing and Communications, Lancaster University. His research interests include computer vision, machine learning, and digital health. He is in the list of Stanford/Elsevier's Global Top 2% Scientists. He received the Best Thesis Award from EEE at NTU and the Best Paper Award, from PREMIA. He has served as the Area Chair for CVPR, ECCV, ICML, NeurIPS, ICLR and MM.